

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## ZPRACOVÁNÍ OBCHODNÍCH DAT FINANČNÍHO TRHU

DIPLOMOVÁ PRÁCE

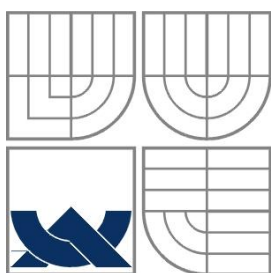
MASTER'S THESIS

AUTOR PRÁCE

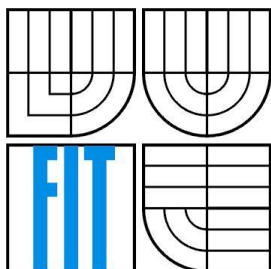
AUTHOR

Bc. TOMÁŠ OLEJNÍK

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# ZPRACOVÁNÍ OBCHODNÍCH DAT FINANČNÍHO TRHU

FOREX DATA PROCESSING

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. TOMÁŠ OLEJNÍK

VEDOUCÍ PRÁCE  
SUPERVISOR

doc. RNDr. JITKA KRESLÍKOVÁ, CSc.

BRNO 2011

## **Abstrakt**

Studijní částí diplomové práce je seznámení se s principy vysoko-frekvenčního obchodování na finančních trzích a primárně se věnuje devizovému trhu. Práce je zaměřena na sběr tržních dat, jejich uložení a následnou filtraci. Rozhodování na základě nekvalitních dat často vede v obchodním světě k fatálním následkům, proto čištění představuje důležitou část při zpracování dat. V práci je popsán adaptabilní algoritmus čištění dat, který je schopen se přizpůsobit jednotlivým trhům a situaci, která na nich v daném momentě panuje. Dle návrhu je implementována modulárně rozšiřitelná aplikace schopná sběru dat, jejich uložení a následné filtrace.

## **Abstract**

The master's thesis' objective is to study basics of high-frequency trading, especially trading at foreign exchange market. Project deals with foreign exchange data preprocessing, fundamentals of market data collecting, data storing and cleaning are discussed. Doing decisions based on poor quality data can lead into fatal consequences in money business therefore data cleaning is necessary. The thesis describes adaptive data cleaning algorithm which is able to adapt current market conditions. According to design a modular plug-in application for data collecting, storing and following cleaning has been implemented.

## **Klíčová slova**

forex, devizový trh, vysoko-frekvenční obchodování, filtrace dat, FIX protokol

## **Keywords**

forex, foreign exchange market, high-frequency trading, data cleaning, FIX protocol

## **Citace**

Olejník Tomáš: Zpracování obchodních dat finančního trhu, Diplomová práce, Brno, FIT VUT v Brně, 2011

# **Zpracování obchodních dat finančního trhu**

## **Prehlásenie**

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením doc. RNDr. Jitky Kreslíkovéj, CSc.

Ďalšie informácie mi poskytol odborný konzultant Michal Kreslík.

Uviedol som všetky literárne pramene a publikácie z ktorých som čerpal.

.....  
Bc. Tomáš Olejník  
25. máj 2011

## **Pod'akovanie**

Týmto by som sa chcel poďakovať doc. RNDr. Jitke Kreslíkovéj, CSc. za vedenie diplomovej práce, za poskytnutie odbornej literatúry a ochotu pri spolupráci.

© Tomáš Olejník, 2011

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

Obsah.....	1
1 Úvod.....	4
2 Finančné trhy .....	5
2.1 Tržné fundamenty.....	5
2.2 Elektronický trh .....	6
2.2.1 Cena .....	6
2.2.2 Charakteristiky trhu .....	7
2.2.3 Príkazy .....	7
2.2.4 Spôsoby obchodovania .....	8
2.2.5 Automatické obchodné systémy .....	8
3 Forex .....	9
3.1 Čo je to forex .....	9
3.2 História devízového trhu.....	9
3.3 Menové páry .....	10
3.4 Obchodovanie na devízovom trhu .....	11
4 Obchodné dáta .....	12
4.1 Zdroje forex dát .....	12
4.2 Zber dát.....	12
4.3 Uloženie dát.....	13
4.3.1 Požiadavky.....	13
4.3.2 Výber úložiska .....	13
4.3.3 Návrh úložiska .....	14
4.4 Špecifikácia požiadaviek .....	15
4.5 Návrh systému .....	15
4.5.1 Knihnica pre zber dát .....	15
4.5.2 Knihnica pre uloženie dát .....	16
4.5.3 Knihnica s filtrom .....	16
5 Filtrácia nekorektných dát.....	17
5.1 Úvod do čistenia dát .....	17
5.2 Dáta a ich chyby .....	18
5.3 Návrh filtra.....	19
5.3.1 Univariantný filter .....	20
5.3.2 Okno pre spracovanie zloženého tiketu.....	20
5.3.3 Okno pre filtráciu skalárnych hodnôt .....	20

5.4	Štatistické charakteristiky .....	20
5.4.1	Vierohodnosť a vernostný kapitál.....	21
5.4.2	Exponenciálne vážený kľzavý priemer .....	22
5.4.3	Hustota tikov.....	23
5.4.4	Očakávaná volatilita a rozptyl .....	23
5.5	Filtračné postupy .....	24
5.5.1	Filtrácia skalárnych hodnôt – úrovňový filter.....	24
5.5.2	Párové filtrovanie.....	25
5.6	Filtračný proces .....	27
5.6.1	Okno pre filtráciu skalárnych hodnôt .....	27
5.6.2	Okno pre spracovanie zloženého tiku.....	33
5.6.3	Univariantný filter .....	34
6	Implementácia.....	36
6.1	Implementačné prostredie.....	36
6.1.1	Technológia Microsoft .NET Framework .....	36
6.1.2	Microsoft SQL Server.....	38
6.1.3	VersaFix.....	38
6.2	Collector - knižnica zberu dát.....	39
6.2.1	FIX protokol .....	39
6.2.2	Hotspot FIX BookFeed API .....	41
6.3	Storage - knižnica pre uloženie dát.....	44
6.3.1	Redundantný zápis.....	44
6.3.2	Modul - zápis do súboru .....	45
6.3.3	Modul – uloženie do databázy .....	45
6.3.4	Synchronizácia databáz.....	45
6.4	Filter.....	47
6.5	Grafické užívateľské rozhranie.....	47
7	Výsledky práce.....	50
7.1	Zber a uloženie obchodných dát .....	50
7.2	Filtrácia obchodných dát.....	50
7.2.1	Ukážkové situácie.....	50
7.2.2	Výsledky čistenia vybranej vzorky dát.....	52
7.3	Čo ďalej? .....	52
7.3.1	Multivariantné filtrovanie.....	53
7.3.2	Zber a filtrácia úplnej knihy objednávok.....	53
8	Záver .....	54
	Literatúra .....	55

Zoznam príloh.....	56
--------------------	----

# 1 Úvod

Ľudstvo sa odjakživa vyvíja. V dávnej minulosti bola perióda jednotlivých pokrokov značne dlhá, no v súčasnej modernej spoločnosti napredujeme priam míľovými krokmi. Kedysi bolo zvykom si zabezpečiť potrebné statky svojpomocne - vypestovať, vyrobiť. Neskôr sa ľudia začali špecializovať na jednotlivé činnosti, začali produkovať viac statkov než sú schopný spotrebovať a vzniká obchod. Pri porovnaní dnešných foriem obchodu s tými historickými by sme veľa podobností nenašli. Avšak samotná podstata obchodu je práve u všetkých foriem rovnaká. Pre všetky zúčastnené strany je cieľom získať z obchodu úžitok.

V tejto diplomovej práci sa budem venovať dnešnému trhu - finančnému trhu. Asi málokto si predstaví tú formu, ktorú mám teraz na mysli. Zaoberať sa budeme vysoko-frekvenčným obchodovaním na medzibankovom devízovom trhu. Detailnejšie sa budeme venovať iba malému zlomku tejto činnosti a tým bude spracovanie dát, ktoré je základným kameňom pre ďalšie obchodné aktivity v tejto oblasti.

Úvodnou požiadavkou bolo naštudovať problematiku obchodovania na finančných trhoch, zamerať sa na obchodné dáta, ich zber, uloženie a prípravu pre použitie vyššími vrstvami. Základná problematika finančných trhov a obchodovanie na nich je diskutované v kapitole 2. Konkrétne zacielenie na devízový trh popisuje kapitola 3, kde sa dozvedáme o histórii trhu, ale aj o jeho špecifikách v porovnaní s ostatnými trhmi. Kapitola 4 sa zameriava na problematiku obchodných dát, kedy postupne od ich popisu prechádzame k ich zberu a uloženiu, kde sú spoločne s diskutovanou problematikou predstavované aj navrhnuté riešenia. Filtrovanie obchodných dát ako dominantnej časti práce sa venujem v kapitole 5, kde je pochytená problematika filtrácie dát a navrhnutý matematicko-štatistický model adaptívneho filtra. V kapitole 6 je podrobnejšie popísaná voľba implementačných prostredí a implementácia špecifikovaných požiadaviek. Zhodnotenie dosiahnutých výsledkov najmä v oblasti čistenia dát zahŕňa kapitola 7.

Práca nadväzuje na semestrálny projekt, ktorý sa zaoberal študijnou a návrhovou časťou aplikácie. Zo semestrálneho projektu sú v upravenej forme použité kapitoly 2 až 4 a úvod kapitoly 5.



## 2 Finančné trhy

Pred samotným riešením problému spracovania obchodných dát, je nevyhnutné vysvetliť základné princípy vysoko-frekvenčného obchodovania. Pochopenie súvislostí a podstaty obchodovania je alfou a omegou ďalšieho štúdia, prípadne experimentov na trhoch. Táto kapitola by mala v jednoduchéj forme zasvätiť čitateľa do problematiky obchodovania. V praxi sú zaužívané niektoré anglické výrazy, ktoré sa zámerne neprekladajú.

### 2.1 Tržné fundamenty

V dnešnej post industriálnej spoločnosti predstavuje obchodovanie hlavný spôsob pre nadobúdanie tovarov a služieb. Možno si to ani neuvedomujeme, ale súčasťou trhu je každý z nás. Z hľadiska ekonómie vieme, že zjednodušené trh je miesto, kde sa stretáva ponuka a dopyt, ktoré vytvárajú cenu. S rastúcou ponukou cena klesá, naopak s rastúcim dopytom rastie aj cena. Tento tržný mechanizmus funguje už od dávnych ľudských období, avšak sa zachoval až do súčasnosti a zatiaľ nebol prekonaný. S rozvojom nových technológií (doprava, komunikačné technológie) sa rapídne mení forma trhov, avšak základný kameň ostáva stále rovnaký.

Fenoménom súčasnosti je elektronické obchodovanie, ktoré prináša svoje výhody a nevýhody. Jednou z hlavných výhod, ktorému sa bude práca významnejšie venovať je rýchlosť realizácie obchodu. Predmetom štúdia obchodovania v diplomovej práci nebude nakupovanie produktov pre vlastnú spotrebu, ale opakovaný nákup a predaj produktov s cieľom dosiahnuť zisk. Cieľom obchodníka nie je držať nakúpené položky, ale nakúpiť lacnejšie a drahšie prediť v čo najkratšom čase (sekundy, minúty alebo maximálne dni). Obchodovanie v krátkych časových intervaloch je možné realizovať vďaka moderným prostriedkom informačných technológií.

Trhu sa účastní veľké množstvo aktérov a jednotlivé obchody ovplyvňujú tržnú cenu. Obchodníci pre obchodovanie využívajú svoje modely, ktoré im signalizujú (prípadne realizujú) jednotlivé obchody. Jedná sa o extrémne náročné štatisticko-matematické systémy, ktoré je nutné neustále vyvíjať, pretože ich funkčnosť veľmi rýchlo expiruje. Dôležitou požiadavkou na model je rýchlosť jeho predikcie a realizácie obchodu, pretože cieľom je realizovať obchod skorej ako ostatní obchodníci, lebo na pomalých už príležitosti nevyšší. Toto obchodovanie sa nazýva vysoko-frekvenčným obchodovaním a systémy, ktoré ho realizujú sa nazývajú vysoko-frekvenčné systémy<sup>1</sup> [1].

---

<sup>1</sup> Zaužívaná je tiež anglická skratka – HFT system (High-frequency trading)

## 2.2 Elektronický trh

Je miesto, kde sa realizujú jednotlivé obchody. Trh je realizovaný pomocou moderných počítačov prepojených pomocou sietí, čo umožňuje ostatným aktérom naň vstupovať. Trh musí zabezpečiť distribúciu cien k jednotlivým účastníkom, ďalej realizáciu jednotlivých obchodov. Príkladom takýchto trhov môže byť trh cenných papierov, komoditný trh alebo devízový trh, ktorému sa budeme venovať v ďalších kapitolách tejto práce. Avšak v tejto fáze abstrahujeme od konkrétnej podoby trhu, vysvetlíme si pojmy ohľadne tržnej ceny a preberieme si základné príkazy, ktoré sa vyskytujú na elektronických trhoch. Tieto základy sú rovnaké pre všetky spomínané trhy. Ak nebude uvedené inak, informácie v tejto časti pochádzajú z [1] a [2].

### 2.2.1 Cena

Každého účastníka trhu isto zaujíma cena, za ktorú je možné daný predmet obchodu predat' alebo kúpiť. Na trhu sa vyskytujú 2 ceny: *bid* cena a *ask*<sup>2</sup> cena (označovať ich budeme ako  $P_{bid}$  a  $P_{ask}$ ). Cenu *bid* charakterizujeme ako cenu dopytu – tržný subjekt je ochotný kúpiť za stanovenú cenu. Opakom je cena *ask*, ktorú charakterizujeme ako cenu ponuky – tržný subjekt ponúka na predaj za stanovenú cenu. Platí  $P_{bid} < P_{ask}$ . U ceny je zobrazené jednotkové množstvo obchodu – koľko jednotiek položky je ochotný obchodník predat' (*ask*) alebo kúpiť (*bid*). Príklad takejto cenovej ponuky môže vyzerat' nasledovne:

$$300/1.02 \times 1.04/500$$

Uvedený údaj reprezentuje želanie kúpiť 300 jednotiek položky za cenu 1.02 ( $P_{bid} = 1.02$ ) a ponuku predat' 500 jednotiek za cenu 1.04 ( $P_{ask} = 1.04$ ). Takéto ponuky posielajú na trh obchodníci, ktorý si želajú obchodovať. Jedno takúto udalosť, ktorá je poslaná na trh nazývame *tikom*.

Na trh prichádza takýchto ponúk veľké množstvo a ich distribúcia všetkým účastníkom by bola veľmi náročná a zároveň nezmyselná, pretože záujmom obchodníka je nakúpiť za čo najnižšiu cenu a predat' za čo najvyššiu a tým maximalizovať zisk. Preto obvykle elektronický trh zostavuje *knihy objednávok* (*order book*), kde jednotlivé ponuky sú zoradené podľa najvyšších (pri *bid*), resp. najnižších (pri *ask*) cien. Príklad knihy objednávok v jednom časovom momente by mohol vyzerat' takto:

$$\dots 700/0.99 \ 500/1.00 \ 300/1.01 \times 1.03|500 \ 1.04|900 \ 1.06|1200\dots$$

Jedná sa len o časť knihy objednávok, kde sú zobrazené 3 najlepšie *bid* ponuky a 3 najlepšie *ask* ponuky. Najlepšiu *bid* ponuku tvorí 1.01 s množstvom 300, ďalej nasledujú 1.00 s množstvom 500 a 0.99 s množstvom 700. Na strane *ask* dominuje 1.03 s množstvom 500, drahší je 1.04 s množstvom

---

<sup>2</sup> často býva označovaná aj ako *offer*

900 a 1.06 s množstvom 1200. Najväčší význam z knihy objednávok má najväčší bid a najmenší ask, často označovaný ako BBO<sup>3</sup>.

Zvláštnym typom tiků, ktorý nie je súčasťou knihy objednávok je *obchod* (angl. *trade*). Týmto nás trh informuje o realizovanom obchode, teda ponuka sa stretne s dopytom a platí  $P_{bid} = P_{ask}$ .

## 2.2.2 Charakteristiky trhu

Najvýznamnejšou charakteristikou trhu je *rozpätie* (angl. *spread*), ktoré vyjadruje rozdiel medzi bid a ask. Spread je cena, ktorú musí platiť obchodník pri realizovaní obchodu, preto je snaha o čo najmenšie rozpätie. Situácia na trhu vyzerá nasledovne:  $P_{bid} = 1.02$  a  $P_{ask} = 1.04$ . V situácii, že sa obchodník rozhodne nakúpiť 100 položiek a následne ich predáť v tom isto čase, je v strate. Pri nákupe zaplatí  $100 \cdot 1.04 = 104$  cenových jednotiek, pričom pri okamžitom odpredaji dostane  $100 \cdot 1.02 = 102$  c.j., čo znamená stratu 2 c.j. (hovoríme, že obchodník platí spread).

*Likvidita* v tržnej terminológii vyjadruje schopnosť nakupovať a predávať s minimálnymi finančnými a časovými nákladmi na realizáciu obchodu. Trh s vysokou likviditou je taký trh, kde jednotliví aktéri nemusia na vybavenie svojich objednávok dlho čakať – vo väčšine prípadov sú vybavené takmer okamžite, čo znamená, že je k dispozícii protistrana na realizáciu obchodu. Obchodník, ktorý na trh prispieva obchodnými ponukami býva označovaný ako poskytovateľ likvidity<sup>4</sup> [1].

*Volatilita* v tržnej terminológii vyjadruje frekvenciu a veľkosť zmeny ceny v čase. Trh s vysokou volatilitou je trh, kde cena často kolíše, je nestála a často mení svoj trend. Opakom je trh, kde cena osciluje okolo určitej hodnoty, prípadne okolo stáleho trendu [6].

## 2.2.3 Príkazy

Obchodovanie na trhu prebieha pomocou príkazov, ktoré naň obchodník posiela. Príkazy *tržnej objednávky* (*market order*), znamenajú obchodovanie na aktuálnej cene. Ak obchodník pošle na trh príkaz typu *limitná objednávka* (*limit order*), znamená to, že si želá realizovať obchod, až cena dosiahne požadovanej hodnoty. Tieto príkazy sa ukladajú do knihy objednávok, naopak príkazy typu *tržná objednávka* akceptujú najlepšiu bid/ask položku, čím sa položka zruší z knihy objednávok. V prípade dosiahnutia požadovanej ceny pri limitnej objednávke, rovnako sa vyberie odpovedajúca položka z knihy objednávok ako pri tržnej objednávke. Obidva druhy objednávok obsahujú minimálne tieto atribúty:

- symbol – označenie nakupovanej položky (napr. menový pár)
- nákup alebo predaj
- množstvo (veľkosť)

---

<sup>3</sup> best bid and offer (ask), top of the book

<sup>4</sup> liquidity provider

- cena

Spôsob akým sú jednotlivé príkazy realizované na konkrétnych trhoch je zväčša komplikovanejší, ale nateraz postačuje len znalosť obecného princípu [4].

## 2.2.4 Spôsoby obchodovania

Samotné obchodovanie môže prebiehať pomocou 2 iniciátorov. Prvým je obchodník ako osoba, kedy na základe svojho vlastného rozhodnutia podporeného technickými signálmi zadá príkaz pre obchod, ktorý by mu mal priniesť zisk. Medzi hlavné výhody tohto obchodovania patrí kontrola obchodníka nad realizovanými príkazmi, ochrana pred neočakávanými stavmi trhu – obchodník dokáže zvyčajne korektne zareagovať. Toto obchodovanie však prináša tiež radu nevýhod ako ľudský stres faktor, pomalosť reakcie, obmedzené množstvo sledovaných trhov súčasne – ľudský mozog nedokáže paralelne spracovať tak rozsiahle informácie ako moderné počítače. Preto väčšina profesionálnejších obchodníkov v súčasnosti preferuje automatické obchodné systémy pre exekúciu svojich obchodných príkazov.

## 2.2.5 Automatické obchodné systémy

Automatické obchodné systémy<sup>5</sup> sú špeciálne počítačové programy (často krát aj špecializované hardvérové komponenty), ktoré realizujú obchody namiesto obchodníka. Tieto systémy obchodujú pomocou množiny naprogramovaných pravidiel, ktoré nazývame obchodné stratégie. Obchodník však nie je len pasívny divák, avšak robota musí neustále kontrolovať, testovať, analyzovať jeho výkonnosť, experimentovať s konfiguráciou, modifikovať a programovať nové stratégie. Podmienky na trhu sa neustále menia, preto robot generujúci zisk dnes môže byť za pár mesiacov úplne nepoužiteľný. Takéto riešenie so sebou prináša kopu výhod, ktorými sú odolnosť proti stresu – program pracuje len podľa naprogramovaných pravidiel, široký záber monitorovaných trhov, rýchlosť identifikácie obchodných príležitostí, rýchlosť a množstvo realizovaných obchodov. Ako každé riešenie problému, tak aj toto má svoje nevýhody. Testovanie stratégií je možné realizovať iba spätne na historických dátach. Nikto nezaručí, že systém generujúci zisk si poradí so všetkými nástrahami reálneho trhu. Vývoj takéhoto systému je finančne náročný, takisto jeho prevádzkové náklady nie sú zanedbateľné. Pozostávajú z nákladov na vývoj a údržbu systému a z nákladov na používanú počítačovú techniku. Rýchlosť je v tomto biznise veľmi dôležitá, preto je dôležité fyzické umiestnenie exekučných serverov, ktoré sa umiestňujú blízko sídla jednotlivých trhov [1].

---

<sup>5</sup> zkratka AOS, často sa používa aj termín *roboti*

## 3 Forex

V diplomovej práci sa budeme venovať podrobnejšie jednému konkrétnemu trhu a tým je medzinárodný menový trh. Prezentované znalosti a princípy budú aplikované priamo na trh forex, avšak zároveň to nepotvrdzuje a ani nevylučuje, že platia aj na iných trhoch. Informácie som čerpal prevažne z [2] a [5].

### 3.1 Čo je to forex

International Interbank **Foreign Exchange** skrátene *forex*, ktorý býva označovaný aj skratkou FX, v našich končinách je skorej známy pod pojmom devízový trh, zjednodušene trh s cudzími menami alebo menový trh. Forex je trh s najväčším obratom, denný obrat sa pohybuje rádovo v biliardoch dolárov a niekoľko násobne prevyšuje obraty na iných finančných trhoch. Forex je decentralizovaný trh (OTC<sup>6</sup>), nemá žiadne centrálné sídlo – burzu, kde by prebiehali všetky obchody. Trh ako celok teda pozostáva z viacerých menších trhov. Čiže trh môžeme chápať ako vzájomne prepojené banky, poisťovne, rôzne fondy, brokerské spoločnosti atď. Tie realizujú medzi sebou obchody – predávajú a kupujú jednotlivé meny. Obchodovanie na trhoch je známe už od dávnejšej minulosti, avšak elektronický trh s cudzími menami v dnešnej podobe je relatívne nová disciplína. Obchodovanie na forex trhu predstavuje v súčasnosti jednu z aktívne veľmi skúmaných oblastí vzhľadom na rozsiahle možnosti informačných technológií, ktoré sa už masívne využívajú, avšak stále majú veľký potenciál vývoja nových technológií a postupov.

### 3.2 História devízového trhu

Korene forex trhu siahajú do hlbšej minulosti, aj keď dnešná podoba trhu cudzích mien sa sformovala len nedávno. V dávnej minulosti bol možný obchod výmenou tovaru za tovar, prípadne služby. Avšak už stovky rokov pred naším letopočtom je zaznamenaná zmienka o používaní odliatkov drahých kovov ako platidla. Jednotlivé mince bolo možné medzi sebou meniť podľa tržnej dohody, čiže výmenný kurz bol ponechaný čisto na dohode predávajúceho a kupujúceho. Hodnota meny bola viazaná na surovinu, obvyčajne na takú, z ktorej bola vyrobená. Tento systém sa postupne vyvíjal a na uvedenom princípe sa zachoval až do 19. storočia. Používanými kovmi boli rôzne zliatiny, ku koncu obdobia najmä striebro a zlato – bimetalický systém.

Koncom 19. storočia nastáva pozvoľný prechod k tzv. zlatému štandardu, čo znamenalo zákonne stanovenú cenu iba zlata. Jednotlivé meny sú kryté zlatom a zároveň zlato slúži ako vodidlo – stanovuje výmenný kurz jednotlivých mien. Po prvej svetovej vojne prišlo k prvému oslabeniu zlata

---

<sup>6</sup> Over-the-counter

nazývanému „štandard zlatej devízy“, ktorý hovorí o zameniteľnosti peňazí za zlato alebo cudziu menu, ktorá je zameniteľná za zlato. Koncom druhej svetovej vojny bola podpísaná tzv. Bretton-Woodska dohoda, ktorá tiež súvisí so založením Medzinárodného menového fondu v roku 1945. Táto dohoda požaduje meniteľnosť zlata iba pre USA a aj to iba pre vonkajšie účely, čo je ďalším oslabením zlata. Boli dohodnuté pevné kurzy jednotlivých mien, pričom priamu väzbu na zlato mal iba americký dolár. Ekonomiky jednotlivých krajín nemali rovnaký rast a taktiež americký dolár menil svoju reálnu hodnotu, preto bola zmena tohto systému len otázkou času. Úplnú väzbu meny na zlato prerušil americký prezident Richard Nixon dňa 15. augusta 1971, kedy odmietol plniť dohodu o predaji zlata bankám za pevnú sadzbu voči americkému doláru. Tento deň sa chápe ako začiatok forex trhu.

Forex od roku 1971 fungoval na princípe pevných výmenných kurzov. Prvými účastníkmi trhu boli svetové banky a veľké investičné spoločnosti. Vo svete panoval stupeň nedôvery voči americkému doláru, ktorého sa začali nadmerne zbavovať a to viedlo v roku 1973 k zrúteniu trhu v tejto podobe. Týmto začala nové obdobie medzinárodného devízového trhu, tak ako ho poznáme v dnešnej podobe, teda na základe variabilných kurzov medzi jednotlivými menami určených tržným mechanizmom.

### 3.3 Menové páry

V predchádzajúcich kapitolách sme jednotky obchodu nazývali všeobecne položky. Forex trh predstavuje výraznú odlišnosť od ostatných trhov, pretože obchodovanie vždy prebieha v pároch položiek, ktoré nazývame menové páry. Jeden menový pár tvorí dvojica mien, pričom závisí na ich poradí. Najvýznamnejšie meny obchodované na forex trhu sú<sup>7</sup>:

- americký dolár (USD)
- euro (EUR)
- japonský jen (JPY)
- britská libra (GBP)
- austrálsky dolár (AUD)
- kanadský dolár (CAD)
- švajčiarsky frank (CHF)

Najviac obchodované menové páry sú:

- euro a japonský jen – EUR/USD
- americký dolár a japonský jen – USD/JPY
- americký dolár a švajčiarsky frank – USD/CHF

---

<sup>7</sup> označenie jednotlivých mien stanovuje norma ISO 4217

- britská libra a americký dolár – GBP/USD.

Prvá z dvojice mien u menového páru sa nazýva základná mena. Druhá sa nazýva kótovacia mena a vyjadruje hodnotu, koľko jednotiek meny je treba na zakúpenie jednej jednotky domácej meny. Napríklad kurz EUR/USD o hodnote 1.3423 vyjadruje cenu jedného eura v dolároch. Menové páry, ktoré neobsahujú americký dolár nazývame krížové páry [2].

### 3.4 Obchodovanie na devízovom trhu

Obchodovanie sa mierne odlišuje od ostatných trhov už spomínaným obchodovaním menových párov. Korektnejšie by sme obchodovanie na forex trhu mali používať termín *výmena*, ktorý vyjadruje výmenu jednej meny za druhú (napr. výmena doláru za euro). Počas jedného obchodu zároveň predávame aj nakupujeme, teda vymieňame. Avšak zaužívané sú pojmy nákup a predaj. Preto je potrebné vždy uvádzať celý menový pár, výraz „*nakúpil som americký dolár*“ má nízku informačnú hodnotu, vždy je dôležité celý menový pár, teda správne „*nakúpil som americký dolár proti euru*“.

Ak momentálne obchodujeme, hovoríme, že sme na pozícii. Rozlišujeme *long* a *short* pozície. Long pozícia vyjadruje nakúpený menový pár, konkrétne nakúpenú základnú menu. Do long pozície sa dostaneme príkazom *buy*, teda akceptovaním ponuky niektorého účastníka trhu. Ak sme v tejto pozícii, očakávame zvýšenie kurzu, aby sme dosiahli zisk. Inverznou pozíciou je *short*, ktorú realizujeme príkazom *sell* a vyjadruje, že daný menový pár sme predali a očakávame pokles jeho kurzu, aby sme boli ziskový pri jeho spätnom nákupe. Na ostatných trhoch predávanie niečoho čo nevlastníme znamená dočasnú výpožičku napríklad od brokerskej spoločnosti. V prípade forex trhu predaj znamená len inverzný nákup, čiže kupujeme *kótovanú* menu [2].

## 4 Obchodné dáta

V diplomovej práci sa budem podrobnejšie venovať zberu forex dát, ich uloženiu a hlavne filtrácii nekorektných dát. Dáta z trhov bývajú často veľmi zašumené a ich čistenie býva podceňované, čo máva fatálne následky pri rozhodovaní. Budovanie obchodných stratégií na zlých základoch, ktorými sú v tomto prípade zašumené dáta, je vopred odsúdené na neúspech.

### 4.1 Zdroje forex dát

Už z predchádzajúcich kapitol vieme, že trh forex je decentralizovaný, čo znamená že zdrojov tržných dát je viacero. Spomedzi zdrojov dát, ktorých spracovanie bude implementovaný trh *HotSpot*<sup>8</sup>. Jedná sa o ucelenú platformu, ktoré umožňuje obchodovanie ako aj poskytuje real-time a historické dáta. Vo všeobecnosti sa pre takéto elektronické finančné trhy používa názov elektronická obchodná sieť (skratka ECN<sup>9</sup>).

Nasledovný popis dát bude konkretizovaný pre dáta poskytované spoločnosťou HotSpot, avšak principiálne je to obdobne aj u iných poskytovateľov dát. Dáta môžu byť poskytované dvomi spôsobmi: *snímky v určitých časových intervaloch* a ako *jednotlivé tiky (udalosti)*.

Spracovanie dát vo formáte snímku je jednoduchšie, pretože snímky obsahujú konkrétnu podobu knihy objednávok v určitom časovom okamžiku. Minimálna doba časového intervalu je 50ms, čo predstavuje pre niektorých obchodníkov nevýhodu, nakoľko môžu dostávať dáta s oneskoreným až do uvádzaného časového intervalu.

Spracovanie jednotlivých tikov je implementačne náročnejšie, pretože si užívateľský program musí aktualizovať svoju internú knihu objednávok. Takisto potrebuje väčšie sieťovú priepustnosť (1 MB/s). Medzi výhody patrí minimálne oneskorenie príchodu informácie, ktorá je generovaná okamžite s jej príchodom na trh, čo môže byť pre niektoré obchodné stratégie rozhodujúce. Rovnako týmto dostávame kompletnú informáciu o trhu.

### 4.2 Zber dát

Trh poskytuje rozhranie pre prístup k trhu, ktoré môže byť štandardizované *protokolom FIX*<sup>10</sup> alebo môže poskytovateľ používať vlastné rozhranie. Táto komunikácia prebieha zasielaním správ sieťou. Služba HotSpotFX využíva pre zasielanie prúdu dát okrem protokolu FIX aj svoj proprietárny protokol, nazvaný *HotSpot ECN ITCH Protokol*<sup>11</sup>.

---

<sup>8</sup> <http://www.hotspotfx.com/>

<sup>9</sup> Electronic Communication Network

<sup>10</sup> <http://www.fixprotocol.org/>

<sup>11</sup> [http://www.hotspotfx.com/pdfs/ITCH\\_Protocol.pdf](http://www.hotspotfx.com/pdfs/ITCH_Protocol.pdf)



Na zber dát bude vytvorený počítačový program, ktorý bude komunikovať s trhom, zasielať a prijímať správy, ktoré bude konvertovať do čitateľnej podoby. Program bude navrhnutý rozšíriteľne, aby bolo možné jednoducho doprogramovať prístup pre ďalšie trhy, resp. jednoducho reflektovať zmeny v aktuálne používanom protokole.

## 4.3 Uloženie dát

Miesto, kde budú dáta fyzicky uložené budeme nazývať *úložiskom*. V nasledujúcich častiach sa budeme venovať jednotlivým spôsobom možnosti riešenia úložiska, výberom konkrétneho spôsobu a návrhom štruktúry.

### 4.3.1 Požiadavky

Na úvod si musíme definovať požiadavky, ktoré musí naše budúce úložisko spĺňať. Dáta budeme využívať prevažne na ich filtráciu, ktorú si popíšeme neskôr. Výstupom filtrácie bude iná štruktúra, čiže pre ďalšie účely použitia dát nemusíme brať ohľad, pretože sa vždy použije už filtrovaný výstup. Hlavnou požiadavkou pre filtráciu historických dát je sekvenčný prístup podobný real-time módu. Ďalej by malo byť jednoduché vyhľadávanie v dátach, najmä vyhľadanie len určitých menových párov, prípadne vymedzenie dát za určité obdobie a vytvorenie rôznych kombinácií. Dáta sú rádovo stovky GB veľké, preto si úložisko musí byť schopné poradiť aj s touto požiadavkou.

### 4.3.2 Výber úložiska

Dáta je možné uchovávať viacerými spôsobmi, medzi ktoré patrí uchovávanie v relačnej databáze, v dátovom sklade a vo formáte textových súborov.

Na úvod začneme *textovými súbormi* uloženými fyzicky na disku. Dáta by boli uložené fyzicky, vo forme textových súborov formátu CSV s oddeľovačom. Mohli by byť pomenované podľa obsahu, prípadne by boli delené do adresárovej štruktúry. Problém s veľkosťou, ani sekvenčným prístupom by nebol, avšak veľký problém nastáva s výberom rôznych kombinácií dát, ktoré by vyžadovalo prístup do viacerých súborov súčasne, prípadne viacpriechodové riešenie.

Ďalšou alternatívou, ktorá na prvý pohľad vyzerá priamo vhodná pre túto úlohu je použitie *dátového skladu*. Dátový sklad je vhodný pre spracovanie veľkého množstva dát a použitie OLAP prostriedkov. Dáta v sklade sa ukladajú do určitých dimenzií, kedy vytvárajú dátovú kocku. Sklad slúži na uloženie subjektovo orientovaných, časovo premenných dát pre podporu rozhodovania. V sklade sú uložené atomické a sumárne dáta [7]. Toto riešenie už vyzerá vhodnejšie, avšak v skutočnosti by bolo vhodné pre uloženie nami vyfiltrovaných dát, pretože dáta vkladané do dátového skladu musia byť čisté a integrované, nakoľko už sa ďalej nemenia a slúžia pre podporu rozhodovania. Rozhodovaním chápeme vytváranie obchodného modelu.

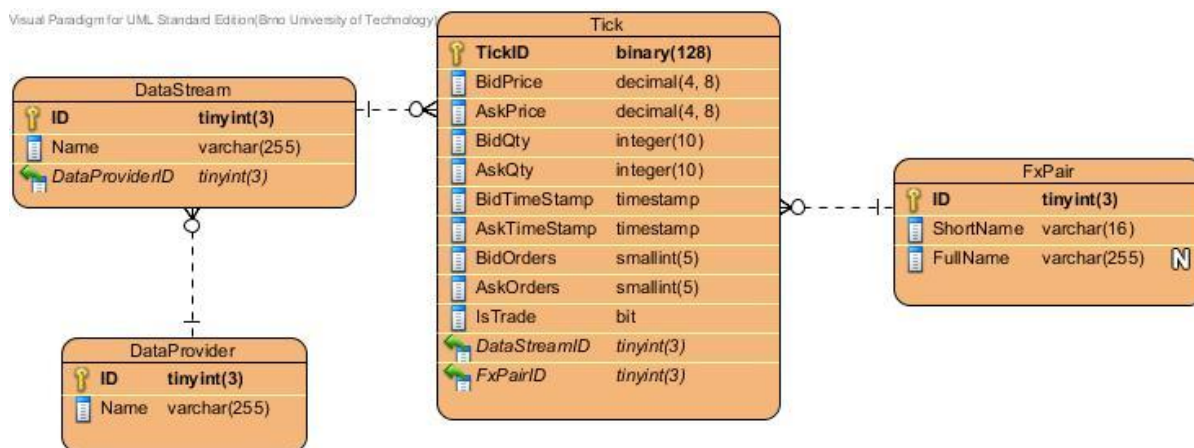
Poslednou zo spomínaných variant je použitie *relačnej databázy*. Pomocou databázy môžeme k dátam pristupovať sekvenčne, selektovať rôzne kombinácie. Taktiež dnešné databázové servery zvládnu uchovávať aj veľké množstvo dát.

Voľba vhodného úložiska padla na relačnú databázu doplnenú o redundantný zápis dát do textového súboru, ktorý umožní ich obnovenie v prípade výpadku pri zápise do databázy.

### 4.3.3 Návrh úložiska

V našom úložisku budeme ukladať jednotlivé tiky z trhu. Jeden tik je snímok vrchu knihy objednávok v čase jej zmeny. Obrázok 4.1 ukazuje konceptuálny návrh databázy. Jednotlivé tiky budú zaznamenávané v tabuľke *Tick*, ktorej atribúty sú:

- *BidPrice* a *AskPrice* – cena bid (ask)
- *BidQty* a *AskQty* – veľkosť ponuky (požiadavky)
- *BidOrders* a *BidQty* – počet ponúk s uvedenou veľkosťou
- *BidTimeStamp* a *AskTimeStamp* – čas, kedy prišiel bid (ask) na trh
- *IsTrade* – identifikuje zrealizovaný obchod
- *TickID* – unikátny 16 bitový identifikátor slúžiaci na jednoznačnú identifikáciu naprieč všetkými používanými úložiskami



Obrázok 4.1: Konceptuálny ER diagram návrhu databázy

Ďalšie tabuľky sú číselníky, ktoré uchovávajú informácie o poskytovateľoch (*DataProvider*), ich prúdov (*DataStream*). Poskytovatelia môžu mať viacero prúdov. Tabuľka *FxPair* uchováva informácie o menovom páre, stĺpec *ShortName* uchováva oficiálnu skratku menového páru a stĺpec *FullName* jeho plný názov, ak existuje.

## 4.4 Špecifikácia požiadaviek

Výsledný systém bude schopný:

- zberať dáta z viacerých forex trhov
- systém bude jednoducho rozšíriteľný o podporu ďalších trhov
- nazberané dáta budú uložené v nefiltrovannej podobe v úložiskách
- implementované budú úložiská typu relačnej databázy a zápisu do súboru
- z úložiska bude možné dáta čítať a predkladať filtru
- výstupom filtrácie bude CSV súbor s prečistenými dátami
- implementovaný filter bude adaptívny a nebude pevne prednastavený pomocou konfigurácie pre konkrétne špecifiká menových párov (podrobnejšie špecifikácia filtra je zachytená v Kapitole 5)
- jednotlivé funkčné časti budú rozdelené do nezávislých knižníc
- ovládanie výsledného systému bude realizované grafickým užívateľským rozhraním, ktoré zapuzdri funkcie poskytované implementované knižnicami

Podrobnejšie požiadavky na systém sú zachytené v predchádzajúcich častiach tejto kapitoly.

## 4.5 Návrh systému

Detailnejší popis návrhu jednotlivých častí už bol spomenutý. V tejto časti sa venujeme celkovému konceptu systému a plánovanému mapovaniu na komponenty. Zo špecifikácie požiadaviek vyplýva, že systém bude plniť 3 hlavné úlohy: zber dát, uloženie dát a ich filtráciu. Na základe týchto funkčných úloh bude výsledné skladať z 3 hlavných častí: knižnica pre zber dát, knižnica s filtrom a databáza.

### 4.5.1 Knižnica pre zber dát

Knižnica pre zber dát sa pripojí k trhu, z ktorého bude prijímať vysielané dáta. Knižnica bude poskytovať výstupné rozhranie reportujúce prijímané tiky z trhu, ktoré bude možné ukladať knižnicou pre uloženie dát. Knižnica sa bude schopná pripojiť k viacerým trhom súčasne a prijímané dáta spojiť do jediného výstupného prúdu tikov. Súčasťou návrhu je aj nezávislosť knižnice od konkrétneho trhu, čo bude riešené jedným z nasledovných spôsobov:

- *Zásuvné moduly* – logiku konkrétneho trhu obsahuje zásuvný modul, ktorý bude možné pripojiť ku knižnici zberu
- *Rozhranie trhu* – knižnica bude implementovať rozhranie, pomocou ktorého bude možné doprogramovať prístupy pre nové trhy. Toto riešenie bude vyžadovať kompiláciu knižnice

## **4.5.2 Knižnica pre uloženie dát**

Knižnica bude prijímať prúd dát posielaný vyššou aplikačnou vrstvou a bude ho ukladať do svojho úložiska. Vyššia vrstva bude posielat' prúd dát získaný z knižnice pre zber dát. Podobne ako knižnica pre zber dát modulárne rozširiteľná o nové úložiska. Možnosti realizácie rozširiteľnosti sú ako v predchádzajúcom prípade.

## **4.5.3 Knižnica s filtrom**

Knižnica bude slúžiť pre filtrovanie dát nazberaných modulom zberu dát alebo dát pochádzajúcich z iných zdrojov. Filter bude implementovaný na základe konceptu prezentovaného v Kapitole 5. Knižnica bude nezávislá na knižnici zberu dát a uloženia dát. Prúd dát bude do knižnice posielaný opäť vyššou aplikačnou vrstvou.

## 5 Filtrácia nekorektných dát

Filtrácia nekorektných dát dominantnou úlohou diplomovej práce, preto sa jej budeme venovať podrobnejšie. Celý koncept filtrácie dát bol prezentovaný v [3], rovnako prevažná väčšina uvedených informácií je čerpaná z tejto publikácie.

### 5.1 Úvod do čistenia dát

Vysoko-frekvenčné dáta sú používané profesionálnymi odborníkmi, ktorí v prípade nekorektných dát vedia správne zareagovať. Horšia situácia nastáva, ak sú dáta spracovávané algoritmicky. Chyby v dátach, hlavne odľahlé hodnoty majú za následok úplne nepoužiteľné výsledky. Podľa výskumov, každé vysoko-frekvenčné dáta obsahujú chybné kóty, ktoré je treba pred algoritmickým spracovaním nevyhnutne odstrániť. Nástroj na čistenie dát nazývame filter. Čistenie je zložitý matematicko-štatistický proces, ktorý býva často podceňovaný. Komplikovaný je z viacerých dôvodov:

- Pre rôznorodosť možných chýb a ich príčin
- Rôznorodosť štatistických vlastností filtrovaných hodnôt
- Rôznorodosť zdrojov dát a ich spoľahlivosti
- Nepravidelnosti v časových intervaloch dát (husté / riedke dáta, dlhé medzery v časových radoch)
- Komplikovanosť kótovaných dát
- Potreba filtrovať dáta v real-time móde (niektoré aplikácie potrebujú výstup okamžite s prichádzajúcimi kótami – nevidíme budúce kóty)

Algoritmus bude adaptívny, bude sa schopný učiť z aktuálne čistených dát. Teraz si vysvetlíme základné pojmy v terminológii filtrovania:

- Príčina vzniku chýb v dátach je väčšinou neznáma. Platnosť danej kóty sa určuje podľa vierohodnosti určenej štatistickými vlastnosťami.
- Susedné kóty sa nazývajú *okno filtra* (*filtering window*) a sú potrebné pre učenie vierohodnosti dát. Okno môže meniť svoju veľkosť v závislosti na filtrovaných dátach.
- Kóty zložitejšej štruktúry (bid-ask kóta) budú rozdelené na jednotlivé skalárne premenné a budú filtrované oddelene. Niektoré typy chýb môžu byť odstránené priamo nad kótami zložitejších štruktúr už pred samotným rozdelením.
- Budeme predchádzať použitiu numerických metód, ktoré vykazujú problémy s konvergenciou, pretože algoritmus musí produkovať stabilné výsledky vo všetkých situáciách.

- Algoritmus musí byť rýchly, sekvenčný a iteratívny. Využije existujúce informácie, keď príde nový tik vypočíta iba minimum potrebných informácií.
- Filter bude schopný práce v 2 módoch: *real-time* a *historický* mód. Kvôli použitiu okna filtra bude obidva módy schopný realizovať jeden filter. V historickom móde bude platnosť kvóty určená až po vysunutí kóty z filtračného okna, naopak v *real-time* bude platnosť určená, hneď po jej príchode.

## 5.2 Dáta a ich chyby

Dátovú chybu charakterizujeme ako kótované dáta, ktoré neodpovedajú reálnej situácii na trhu. Dátovou chybou je, ak je zle uvedená cena alebo cena neodpovedá času, kedy bola reportovaná. Existuje veľa dôvodov vzniku chýb, ktoré sa dajú rozdeliť do 2 skupín:

1. *Ľudské chyby*: Chyby spôsobené zlyhaním ľudského faktora
  - Neúmyselné chyby (preklepy)
  - Úmyselné chyby (rôzne testovacie tiky)
2. *Systémové chyby*: Chyby spôsobené počítačovými systémami

V mnohých prípadoch je ťažké nájsť presnú príčinu vzniku chyby. Cieľom filtra je identifikovať chybu bez ohľadu na dôvod jej vzniku. Niekedy môže byť príčina vzniku chyby identifikovaná pomocou vlastností chybných tikov.

V dátach sa vyskytujú nasledovné chyby:

1. *Desatinné chyby*: Chyba vzniká pri zmene väčšieho desatinného čísla kótovanej ceny. Napríklad aktuálna kóta je 1.3498, nasledovaná správnou kótou 1.3501, avšak kóta uverejnená na trh je 1.3401. Táto chyba najviac poškodzuje algoritmy, ktoré si uchovávajú kvóty v dočasnej pamäti, ktorá tam môže ostať po veľmi dlhú dobu a spôsobiť sériu chybných kót.
2. *Test kóty*: Niektorí účastníci trhu posielajú testovacie tiky na trh v čase jeho nízkej likvidity. Tieto tiky môžu spôsobiť veľké škody, pretože pre filter môžu vyzerieť hodnoverne. Uvedieme si 2 príklady:
  - *Skorý ranný test*: Účastník trhu posielal na trh neplatný tik s cieľom zistiť, či je spojenie s trhom funkčné. Žiadny ľudský obchodník neberie prvé tiky vážne, ale filter by sa mohol nechať zmiasť. Preto filter musí byť veľmi ostražitý k prvým tikom, ktoré prídu po dlhej dátovej medzere.
  - *Monotónne série tikov*: Niektorí účastníci zvyknú posielat' dlhé série konštantne rastúcich alebo klesajúcich tikov s cieľom merania odozvy trhu. Tieto série

z pohľadu štatistických metód vyzerajú veľmi vierohodne, jediné na základe čoho ich môže filter identifikovať, je ich monotónne správanie.

3. *Opakované tiky*: Skupina účastníkov trhu zvykne svoje posledné tiky vysielat' opakovane v určitých časových intervaloch. Toto je neškodné, pokiaľ sa to deje v rozumnej miere. Na niektorých trhoch je to dokonca obvyklý jav. Problém nastáva, ak takto opakuje tiky tisíc násobne s vysokou frekvenciou, kedy sa môžu strácať dôležité tiky iných účastníkov.
4. *Kopírovanie tikov*: Niektorí účastníci používajú svoje počítače pre kopírovanie tikov iných účastníkov. Pokiaľ tieto dáta sú korektné sa nič zásadného nedeje. Problém nastáva ak sú tieto dáta mierne modifikované, pretože filter dokáže identifikovať monotónne alebo opakujúce sa sekvencie dát, avšak tieto dáta nespĺňajú ani jedno kritérium.
5. *Problém mierky*: Kótovacie konvencie sa môžu na určitých trhoch odlišovať. Jeden poskytovateľ dát môže kótovať danú menu pre počet 100 jednotiek, iný len pre 1 jednotku.

## 5.3 Návrh filtra

Návrh filtra vychádza tiež z [3]. Filter bude vykonávať nasledovné operácie:

1. Bude prijímať finančné tiky zoradené podľa ich času príchodu na trh
2. Tieto finančné tiky bude reportovať zoradené podľa ich časov a budú prefiltrované. Pre každý tik filter pridá nasledovné hodnoty:
  - *Vierohodnosť tiky*, prípadne jeho jednotlivých zložiek (bid, ask a rozpätie) bude vyjadrená reálnym číslom z intervalu  $[0;1]$ , pričom 0 vyjadruje úplne neplatný tik a 1 vyjadruje úplne platný tik
  - *Platnosť tiky*, označuje tik ako platný alebo neplatný na základe stanovenej prahovej vierohodnosti

Bežný užívateľia budú používať tiky, ktoré budú mať vierohodnosť väčšiu ako nastavená prahová hodnota (často používaná je 0.5). Pre real-time operácie sa bude používať výstup aktuálneho tiky, ktorý filter spracuje a naopak pre historické budú použité tiky, ktoré budú upravené až po určitom počte ďalších novších tikov. Historické filtrovanie bude teda presnejšie.

Pred samotným spustením filtra na reálne dáta filter potrebuje určité *adaptačné obdobie*, čo znamená že filter pobeží na dátach určité časové obdobie, čím si nastaví svoje štatistické interné hodnoty pre potreby konkrétneho trhu (adaptuje sa).

Hierarchiu čistiaceho algoritmu rozdelíme do nasledovných úrovní (od najvyššej po najnižšiu).

### 5.3.1 Univariantný filter

Filter najvyššej úrovne, ktorý obsluhuje filtráciu celej jednej série. Sériou rozumieme jeden menový pár – od toho názov univariantný. Jeho úlohou je posielanie prichádzajúcich tikov do nižších úrovní hierarchie filtra, ako aj zbieranie výsledkov reportovaných z nich reportovaných a ich transformácia do výstupného formátu. Realizuje real-time a historické filtrovanie. Taktiež podporuje niektoré filtračné operácie, ktoré je možné realizovať na tejto úrovni a nižším vrstvám už môže posilať čiastočne korigované dáta.

### 5.3.2 Okno pre spracovanie zloženého tikku

Táto vrstva filtra sa nachádza na strednej úrovni hierarchie vrstiev. Obsahuje sekvenciu posledných zložených tikov (bid a ask hodnoty), niektoré z nich už môžu byť korigované nadradenou vrstvou. Úlohou tejto vrstvy je rozdelenie zložených tikov na skalárne hodnoty a poslanie ich do prislúchajúcich okien nižšej vrstvy. Taktiež testuje hodnoty na jednoducho identifikovateľné chyby (hodnoty mimo doménu – záporné ceny ...). Nad danými dátami vykoná požadované matematické transformácie (napr. vyráta logaritmus ceny). Všetky tieto operácie realizuje ešte nad zloženými tikmi (bid a ask súčasne). Takisto prijíma výsledky filtrovania nižšej vrstvy a spracúva ich. Ak sú kóty finálne prefiltrované, posila ich do nadradenej vrstvy.

### 5.3.3 Okno pre filtráciu skalárnych hodnôt

Táto najnižšia vrstva filtra obsahuje sekvenciu posledných skalárnych kót, ktoré sú stále upravované. Jeden filter obsahuje 3 skalárne okná (bid, ask a rozpätie). Úlohou tejto vrstvy je testovanie novo prijatých skalárnych hodnôt, ich porovnávanie s ostatnými hodnotami v aktuálnom okne filtra. Vyrátame hodnotu vierohodnosti aktuálnej hodnoty, na základe nej prerátame hodnoty vierohodnosti ostatných kót v okne. Ak už je hodnota niektorého z posledných (najstarších) kót v okne finálna, kótu z okna vyradíme a reportujeme ju do vyšších vrstiev. Pomocou korektných kót aktualizujeme interné sledované štatistické charakteristiky filtra, čo zabezpečuje jeho stálu adaptabilitu.

## 5.4 Štatistické charakteristiky

Počas filtračného procesu si uchováame štatistické charakteristiky pre podporu rozhodovania a adaptabilitu filtra. V tejto podkapitole budú postupne rozobraté jednotlivé štatistiky vrátane ich matematických definícií. Sledované štatistiky tiež nazývame on-line štatistiky, pretože ich hodnota sa pravidelne aktualizujú podľa prichádzajúcich dát z trhu. Štatistiky sú prevzaté z [3].



### 5.4.1 Vierohodnosť a vernostný kapitál

Na úvod potrebujeme vysvetliť používané veličiny. Prvou z nich je *vierohodnosť* (angl. *credibility*), ktorá je primárnym výstupom filtračného algoritmu. Označovať ju budeme  $C$ . Nadobúda hodnôt od 0 do 1, pričom 0 značí úplne neplatný tik a 1 značí úplne platný tik. Túto hodnotu by sme mohli vyjadriť ako pravdepodobnosť toho, že daný tik je platný. Z dôvodu veľkej rozmanitosti tikov na trhu nedokážeme s istotou určiť platnosť tik – jedná sa o platnosť v zmysle fuzzy logiky. Vierohodnosť nie je aditívna, vierohodnosť tik meraného viacerými individuálnymi testami nie je sumou ich výsledných vierohodností. Vyplýva to z oboru hodnôt vierohodnosti, kedy by sme sa sčítaním hodnôt napríklad 0.6 a 0.8 dostali mimo obor hodnôt.

Potreba aditívnej veličiny pre vnútorné výpočty počas filtrácie vedie k zavedeniu novej aditívnej veličiny. Definujeme si aditívny *vernostný kapitál* (angl. *trust capital*), ktorý budeme označovať  $T$ . Vernostný kapitál je definovaný v celom rozsahu reálnych čísiel. Úplne platná hodnota náleží  $T = \infty$  a úplne neplatná  $T = -\infty$ . Zavedieme monotónnu reláciu na konverziu medzi vierohodnosťou a vernostným kapitálom

$$C(T) = \frac{1}{2} + \frac{T}{2\sqrt{1+T^2}}$$

Rovnica 5.1: Prevod vernostného kapitálu na vierohodnosť

a inverznú reláciu

$$T(C) = \frac{C - \frac{1}{2}}{\sqrt{C(1-C)}}$$

Rovnica 5.2: Prevod vierohodnosti na vernostný kapitál

Vernostný kapitál sme definovali ako aditívnu veličinu a nakoľko máme definovanú reláciu medzi vernostným kapitálom a vierohodnosťou (Rovnica 5.1 a Rovnica 5.2), dokážeme definovať operáciu sčítania aj pre vierohodnosť

$$C_{sum} = C \left[ \sum_{i=0}^n T(C_i) \right]$$

Rovnica 5.3: Aditívna operácia dvoch vierohodností pomocou vernostného kapitálu

Uvedenú operáciu je možné realizovať pre ľubovoľný počet vierohodností, avšak musia byť na sebe navzájom nezávislé. Výslednú vierohodnosť dvoch nezávislých vierohodností znázorňuje Tabuľka 5.1. Hodnoty (0,5) v zátvorkách uvádzajú nedefinované výsledky, kedy výsledná vierohodnosť môže nadobúdať hodnoty v rozmedzí 0 až 1

<b>C<sub>sum</sub></b> $C_1 =$	<b>0</b>	<b>0,25</b>	<b>0,5</b>	<b>0,75</b>	<b>1</b>
$C_2 =$					
<b>1</b>	(0,5)	1	1	1	1
<b>0,75</b>	0	0,5	0,75	0,878	1
<b>0,5</b>	0	0,25	0,5	0,75	1
<b>0,25</b>	0	0,122	0,25	0,5	1
<b>0</b>	0	0	0	0	(0,5)

Tabuľka 5.1: Celková vierohodnosť vyrátaná z dvoch nezávislých vierohodností

## 5.4.2 Exponenciálne vážený kľzavý priemer

Základným kameňom, na ktorom budeme stavať ďalšie sofistikovanejšie komponenty filtra je exponenciálne vážený kľzavý priemer (angl. exponential moving average – EMA). Jednoduchý kľzavý priemer slúži na odhaľovanie trendov v časových radoch, je to aritmetický priemer hodnôt za vopred stanovené časové obdobie. My budeme používať jeho exponenciálnu variantu, pri ktorej sú novo pridané hodnoty váhované pomocou rozpadajúceho sa jadra (angl. kernel):

$$ema(t) = \frac{e^{-t/\tau}}{\tau}$$

Rovnica 5.4: Jadro exponenciálne váženého kľzavého priemeru

Výpočet exponenciálne váženého kľzavého priemeru je veľmi efektívny, pretože jeho rekurzívny zápis sa dá realizovať pomocou jednoduchého iteratívneho výpočtu (Rovnica 5.5). Celá konvolúcia sa prakticky nikdy nepočíta, pre každý nový tik je realizovaných len pár sčítaní a násobení.

$$EMA[\tau; z](t_n) = \mu EMA[\tau; z](t_{n-1}) + (v - \mu)z_{n-1} + (1 - v)z_n$$

kde

$$\alpha = t_n - t_{n-1}$$

$$\mu = e^{-\alpha}$$

$$v = \begin{cases} 1, & \text{interpolácia predchádzajúcim bodom} \\ \frac{(1 - \mu)}{\alpha}, & \text{lineárna interpolácia} \\ \mu, & \text{interpolácia aktuálnym bodom} \end{cases}$$

Rovnica 5.5: Iteratívny výpočet exponenciálne váženého kľzavého priemeru

### 5.4.3 Hustota tikov

*Hustota tikov* (angl. *quote density*) vyjadruje množstvo tikov v poslednom časovom intervale. Výpočet kľzavého priemeru prebieha iteratívnym spôsobom s použitím varianty interpolácie aktuálnym bodom, ako bolo uvedené v predchádzajúcej kapitole.  $\Delta\vartheta_r$  je nastaviteľný rozsah jadra kľzavého priemeru. Premenná  $c_d$  je váha tikú, ktorá obvykle nadobúda hodnotu 1 a znižuje sa len v prípade opakovaných tikov s rovnakou hodnotou, ktoré majú pre nás menšiu informačnú hodnotu. Pri zmene sledovanej hodnoty sa premenná  $c_d$  nastavuje znova na hodnotu 1.  $\delta\vartheta$  je časový rozdiel medzi aktuálnym a predchádzajúcim tikom.

$$d = EMA \left[ \Delta\vartheta_r; \frac{c_d}{\delta\vartheta} \right]$$

Rovnica 5.6: Hustota tikov

### 5.4.4 Očakávaná volatilita a rozptyl

Očakávaná volatilita je štatistická charakteristika, ktorá vyjadruje očakávanú premenlivosť trhu na základe nedávnych vlastností trhu. Algoritmus je časovo náročnejší, preto je vhodné do výpočtu zaradiť len platné tiky.

Na výpočet budeme potrebovať pomocnú štatistickú charakteristiku trhu, ktorú nazveme „mikro“-volatilita. Definovaná je ako kľzavý priemer druhej mocniny filtrovanej premennej pomerenej k zmene času medzi dvomi sledovanými tikmi.  $\Delta\vartheta_r$  je opäť nastaviteľný rozsah jadra kľzavého priemeru.

$$v = EMA \left[ \Delta\vartheta_r; \frac{(\delta x)^2}{\delta\vartheta} \right]$$

Rovnica 5.7: Výpočet "mikro"-volatility

Výsledná očakávaná volatilita  $v_{exp}$  bude vypočítaná ako maximum z 3 „mikro“-volatilit:  $v_{fast}$ ,  $v$  a  $v_{slow}$ , ktoré sa budú líšiť nastavením hodnoty rozsahu jadra kľzavého priemeru  $\Delta\vartheta_r$ . Najkratší rozsah bude mať  $v_{fast}$ , naopak najdlhší rozsah bude mať  $v_{slow}$ . Použitie len jedného rozsahu jadra nie je možné z dôvodu premenlivosti trhu. V prípade extrémnych udalostí na trhu  $v_{fast}$  umožní filtru sa rýchlo prispôsobiť novovzniknutej situácii. Naopak  $v_{slow}$  zabráni zahodeniu tikov v málo likvidných hodinách.

$$v_{exp} = \max(v_{fast}, v, v_{slow})$$

Rovnica 5.8: Očakávaná volatilita

Filter ďalej potrebuje špeciálny nástroj na zabránenie prijatia zmien hodnôt počas dlhých časových medzier bez tikov. Dátové medzery sú charakterizované veľkým  $\delta\vartheta$  a veľmi malým množstvom tikov. Požadovanú úpravu dosiahneme použitím nasledovného výrazu:

$$\Delta\vartheta_{corr} = \min \left[ \frac{2,5Q}{d}, \max \left( \frac{0,1Q}{d}, \Delta\vartheta \right) \right]$$

Rovnica 5.9: Upravená rozdiel času

kde  $d$  hustota tikov (Rovnica 5.6) a  $Q$  je počet platných tikov v danom intervale (Rovnica 5.29). Rovnica 5.9 tiež nastavuje dolný časový limit  $\Delta\vartheta_{corr}$  v prípade vysokej frekvencie platných dát.

Upravenú hodnotu  $\Delta\vartheta_{corr}$  použijeme pre výpočet očakávaného rozptylu (strednej kvadratickej odchýlky):

$$V(\Delta\vartheta_{corr}) = \Delta\vartheta_{corr} v_{exp}$$

Rovnica 5.10: Očakávaný rozptyl

## 5.5 Filtračné postupy

Filter je postavený na 2 základných postupoch vyhodnocovania hodnôt, ktoré sú do značnej miery odlišné, avšak obidve sú pre korektnú filtráciu potrebné.

### 5.5.1 Filtrácia skalárnych hodnôt – úrovňový filter

Filter skalárnych hodnôt je jedna z častí filtra, ktorá sa zaoberá filtrovaním prúdu hodnôt. Tento filter slúži pre úvodný výpočet vierohodnosti filtrovanej premennej. Tento filter sa aplikuje len na tie časové rady, u ktorých nie je dôležitá len zmena hodnoty, ale aj samotná hodnota má význam. V našom prípade tento filter aplikujeme na bid-ask rozpätie, ktoré môže byť premenlivé z tik na tik, avšak jeho hodnota ostáva v určitom pevnom rozsahu, ktorý sa mení iba v relatívne dlhých časových intervaloch (mesiace, roky). Od toho aj názov *úrovňový*, pretože hodnota zostáva dlhodobo na určitej hladine – úrovni.

Pre výpočet budeme potrebovať priemernú hodnotu sledovanej veličiny, ktorú vyrátame pomocou iteratívneho výpočtu kľzavého priemeru:

$$\bar{x} = EMA[\Delta\vartheta_r; x]$$

Rovnica 5.11: Kľzavý priemer sledovanej hodnoty

$$\hat{x} = \frac{x - \bar{x}}{\sqrt{EMA[\Delta\vartheta_r; (x - \bar{x})^2]}}$$

Rovnica 5.12: Pomer priemeru hodnoty a jej štandardnej odchýlky

Ďalšou potrebnou hodnotou je podiel priemeru a štandardnej odchýlky (Rovnica 5.12), ktorý budeme označovať  $\hat{x}$ .  $\Delta\vartheta_r$  je opäť nastaviteľný rozsah jadra kľzavého priemeru.

Nízka hodnota  $|\hat{x}|$  značí veľkú dôveru, naopak extrémne veľká hodnota  $|\hat{x}|$  znamená odľahlý tik, čo je tik s nízkou vierohodnosťou a záporným vernostným kapitálom. Definujeme si základný vernostný kapitál, ktorý bude pomerný k  $\hat{x}^2$ . Pre odľahlé hodnoty bude základný vernostný kapitál

záporný, ale pre malé hodnoty bude kladný, pričom maximálna hodnota bude 1. Výsledná definícia základného vernostného kapitálu  $i$ -tej kóty prichádzajúcej do úrovňového filtra je:

$$T_{i0} = 1 - \left( \frac{\hat{x}_i}{\varepsilon_0} \right)^2$$

Rovnica 5.13: Základný vernostný kapitál  $i$ -tej kóty úrovňového filtra

kde 0 značí, že sa jedná o výstup úrovňového filtra a  $\varepsilon_0$  je konštanta slúžiaca na zmenšenie hodnoty  $\hat{x}_i$ . Hodnota konštanty  $\varepsilon_0$  bola stanovená experimentálne na 5,5. Rovnica 5.13 vyjadruje jednoduchý spôsob ako dosiahnuť požadované správanie. Pre potreby forex trhu je táto verzia úrovňového filtra postačujúca.

## 5.5.2 Párové filtrovanie

Porovnanie tikov v pároch je najdôležitejšou operáciou výsledného adaptívneho filtra, preto sa jej budeme venovať podrobnejšie. Pri tejto operácii porovnávame tiky, ktoré nie sú len susediace, ale nachádzajú sa v celom okne filtra. Na rozdiel od úrovňového filtra je párové filtrovanie aplikované na všetky typy hodnôt, ktoré prichádzajú do filtra – bid, ask a rozpätie.

Dôležitou súčasťou je filter zmeny, ktorého úlohou je posúdiť vierohodnosť zmeny sledovanej premennej medzi dvomi ľubovoľnými kótami (nachádzajúcimi sa v okne, prípadne sa do neho chystajúcimi vstúpiť). Významným parametrom je časová vzdialenosť 2 kót – všeobecne pri dlhšom časovom úseku sa dá predpokladať väčší rozdiel sledovanej hodnoty. Dôležitým kritériom je adaptabilita, preto budú použité kľzavé on-line štatistické charakteristiky, ktoré sú prichádzajúcimi tikmi modifikované.

Zmena filtrovanej premennej  $x$  medzi  $j$ -tou a  $i$ -tou kótou je definovaná ako

$$\Delta x_{ij} = x_i - x_j$$

Rovnica 5.14: Zmena sledovanej premennej

kde  $x_i$  a  $x_j$  sú transformované hodnoty v zmysle Kapitoly 5.6.2.2 na strane 33.

Relatívna zmena sledovanej premennej je potom definovaná ako:

$$\varepsilon_{ij} = \frac{\Delta x_{ij}}{\varepsilon_0 \sqrt{V(\Delta \vartheta_{ij})}}$$

Rovnica 5.15: Relatívna zmena sledovaného páru hodnôt

Očakávaný rozptyl  $V(\Delta \vartheta)$  je vypočítaný pomocou on-line štatistickej charakteristiky (Rovnica 5.10).  $\varepsilon_0$  je malá kladná konštanta, ktorej hodnota bola experimentálne určená na 5,5. Nízka hodnota  $|\varepsilon|$  znamená veľkú dôveru danému páru hodnôt – pravdepodobne obidve hodnoty sú korektné, naopak extrémne vysoká hodnota  $|\varepsilon|$  značí nízku vierohodnosť a záporný vernostný kapitál, čo znamená, že jedna z dvoch hodnôt bude odľahlou.

Vernostný kapitál budeme označovať prívlastkom čiastkový. Výpočet je obdobný ako u úrovňového filtra. V tomto prípade budeme tento čiastkový vernostný kapitál označovať  $U$ , aby

sme ho odlišili od celkového vernostného kapitálu, ktorý závisí od viacerých parametrov. Čiastkový vernostný kapitál je závislý na  $\varepsilon_{ij}^2$ :

$$U_{ij} = U(\varepsilon_{ij}^2) = 1 - \varepsilon_{ij}^2$$

Rovnica 5.16: Čiastkový vernostný kapitál sledovaného páru hodnôt

Tento čiastkový vernostný kapitál bude použitý na modifikáciu celkového vernostného kapitálu  $T_i$  skúmanej kóty, čo bude vysvetlené neskôr. Hodnota  $\varepsilon_{ij}$  rovná 1 znamená čiastkový vernostný kapitál rovný 0, teda nezvyšuje ani neznižuje celkový vernostný kapitál sledovanej kóty. Malá hodnota  $\varepsilon_{ij}$  vedie ku kladnému vernostnému kapitálu  $U_{ij}$  – zvyšuje celkový vernostný kapitál, veľké hodnoty znamenajú záporný vernostný kapitál  $U_{ij}$ , čo vedie k zmenšovaniu celkového vernostného kapitálu sledovanej premennej.

Prezentovaný vzorec je vhodný pre porovnávanie nezávislých kót oddelených malými časovými intervalmi. Avšak treba zobrať do úvahy 2 nasledujúce situácie, ktoré vznikajú pri párovom filtrovaní:

1. Filtrovanie sa musí držať časovej osi aj v prípade, že kóty majú zopár blízkych a väčšie množstvo vzdialených kót. Pokiaľ budeme celkový vernostný kapitál počítať aditívnym spôsobom (Kapitola 5.6.1.1), musíme zabezpečiť, aby časovo vzdialené kóty mali oveľa menšiu váhu ako blízke kóty a neovplyvnili tak negatívne výsledok filtrovania. Toto dosiahneme definovaním vernostného kapitálu závislom na  $(\Delta\vartheta)^{-3}$ .
2. Pre veľké hodnoty  $\Delta\vartheta$  môžu byť odľahlé hodnoty príliš ľahko prijaté (Rovnica 5.16). Preto je dôležité, aby pozitívna hodnota vernostného kapitálu sa s rastúcim  $\Delta\vartheta$  znižovala len veľmi mierne, naopak v prípade záporného vernostného kapitálu musí jeho hodnota zostať nízka (nesmie sa blížiť k nule) aj pre veľké  $\Delta\vartheta$ . Filter teda potrebuje rozdeliť rýchlosť úpravy celkového vernostného kapitálu:

- Rýchlo pre malé hodnoty  $\varepsilon$  – kladný vernostný kapitál
- Pomaly pre veľké hodnoty  $\varepsilon$  – záporný vernostný kapitál

Táto úprava je dôležitá pre dlhé časové úseky bez tikov.

Pri filtrácii treba vychádzať z predpokladu, že v prípade ťažkého rozhodnutia o vierohodnosti kóty, je jej odmietnutie prijateľnejšie ako jej prijatie. Avšak toto je možné urobiť len v extrémnych prípadoch, pretože odmietnutých kót nesmie byť veľký počet (pokiaľ sa samozrejme nejedná o extrémne zašumené dáta).

Aplikáciou vyššie uvedených podmienok nám vzniká rozšírená výsledná rovnica pre výpočet párového vernostného kapitálu:

$$T_{ij} = T(\varepsilon_{ij}^2, \Delta\vartheta_{ij}) = \frac{1 - \varepsilon_{ij}^4}{1 + \varepsilon_{ij}^2 + \left(\frac{d \Delta\vartheta_{ij}}{v}\right)^3}$$

Rovnica 5.17: Výpočet párového vernostného kapitálu

Neznámá  $d$  je hustota tikov (Rovnica 5.6), konštantná hodnota  $v$  odpovedá obvyklému časovému intervalu medzi kótami (približne  $1/d$ ).

Tabuľka 5.2 zobrazuje správanie hodnoty vernostného kapitálu párového filtrovania závislého na hodnotách výrazov  $|\varepsilon|$  a  $d \Delta\theta/v$  dvoch kót. Vernostný kapitál sa približuje o nule oveľa rýchlejšie so zväčšujúcou sa časovou vzdialenosťou pre malé hodnoty  $|\varepsilon|$  ako pre väčšie.

<b>T</b> $\frac{d \Delta\theta}{v} =$ $ \varepsilon  =$	<b>0</b>	<b>0,5</b>	<b>1</b>	<b>2</b>	<b>4</b>
<b>4</b>	-15,0	-14,9	-14,2	-10,2	-3,2
<b>2</b>	-3,0	-2,9	-2,5	-1,2	-0,22
<b>1</b>	0	0	0	0	0
<b>0,5</b>	0,75	0,68	0,42	0,10	0,014
<b>0</b>	1	0,89	0,50	0,11	0,015

Tabuľka 5.2: Vernostný kapitál T párového filtrovania

## 5.6 Filtračný proces

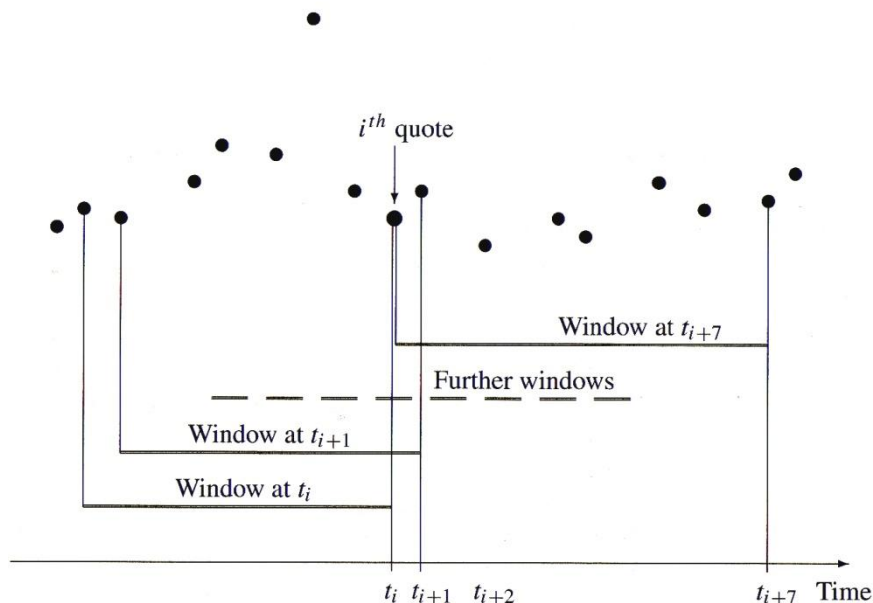
V tejto časti bude popísaný princíp aplikácie jednotlivých štatistík a definovaných výpočtov na prichádzajúce tiky, rozposielanie tikov vyššími vrstvami do nižších vrstiev filtra, kde budú spracované a následne reportované znova do vyšších vrstiev.

### 5.6.1 Okno pre filtráciu skalárnych hodnôt

Okno pre filtráciu skalárnych hodnôt sa nachádza na najnižšej úrovni hierarchie filtra. Obsahuje množinu posledných kót zoradených podľa času. Tieto kóty slúžia na určovanie vierohodnosti novo prichádzajúcej kóty. Novo prichádzajúce kóty sa vkladajú na koniec okna a okno opúšťa kóty zo začiatku okna na základe výstupnej rovnice (Rovnica 5.30). Okno sa mení v čase a jeho veľkosť nie je konštantná.

Všetky skalárne kóty nachádzajúce sa v okne majú nastavené dočasné hodnoty vierohodnosti, ktoré sú vždy príchodom novej kóty modifikované. V prípade ak kóta opustí okno, hodnota jej vierohodnosti sa stáva konečnou. Hodnoty, ktoré prekročia nastavený prah vierohodnosti označujeme ako vierohodné kóty. Tieto kóty pri výstupe z okna aktualizujú hodnoty štatistickej on-line analýzy potrebné pre zabezpečenie adaptability filtra.

Na začiatku filtrovania je okno prázdne, preto na prvú hodnotu nie je možné aplikovať párové filtrovanie – aplikujeme iba úrovňový filter. Príklad skalárneho okna počas filtrovania znázorňuje Obrázok 5.1.



Obrázok 5.1: Skalárne okno počas filtrovania (prevzaté z [3])

#### 5.6.1.1 Príchod novej kóty do okna

Do okna prichádzajú len tie kóty, ktoré prejdú základným overovaním testom vyšších vrstiev. Pri príchode novej kóty do okna je vykonaný potrebná analýza, ktorá je prevedená iteratívnym spôsobom a využíva aktuálne hodnoty on-line štatistík pre posúdenie kóty. Na tejto úrovni filtra pracujeme s normalizovanými hodnotami ako bolo vysvetlené v Kapitole 5.6.2.2, ktoré sú k dispozícii z nadradených vrstiev. Nasledujúce výpočty sú realizované pri príchode novej kóty:

1. V prípade kóty typu bid-ask rozpätie, tak vyrátame základný vernostný kapitál  $T_{i0}$  (Rovnica 5.13). Tento výsledok je ešte vynásobený konštantou  $c_{level}$ , ktorá vyjadruje dôraz kladený na úrovňový filter. V prípade hodnoty reprezentujúcej cenu je  $T_{i0} = 0$ .
2. Novo prichádzajúca hodnota je porovnaná s ostatnými hodnotami pomocou párového filtrovania (Kapitola 5.5.2). Vyrátaný párový vernostný kapitál  $T_{ij}$  (Rovnica 5.17) ovplyvňuje celkový vernostný kapitál prichádzajúcej kóty  $T_i$  a tiež aktualizuje vernostný kapitál starších kót  $T_j$  nachádzajúcich sa v okne.

Pre výpočet  $T_{ij}$  potrebujeme hodnotu očakávaného rozptylu (Rovnica 5.10) a upravený časový rozdiel medzi hodnotami (Rovnica 5.9), kde potrebujeme hodnotu  $Q$  prichádzajúcej kóty, ktorú vyrátame na základe známej hodnoty  $Q_j$  staršej kóty:

$$Q = Q_j + 1$$

Rovnica 5.18: Výpočet  $Q$  pre novo prichádzajúcu kótu

Vernostný kapitál novej kóty je vypočítame na základe vzťahu:



$$T'_i = c_{level} T_{i0} + \sum_{j=i-n}^{i-1} C_j T_{ij}$$

Rovnica 5.19: Výpočet normálneho vernostného kapitálu pomocou párového filtrovania a úrovňového filtra

Označenie  $T'_i$  značí, že sa nejedná o konečný vernostný kapitál – v niektorých prípadoch môže byť ešte modifikovaný. Vo vzťahu k úrovňovému filtru prirátame vážený súčet aktuálnych vierohodností kót nachádzajúcich sa v okne vynásobený výsledkom párového filtrovania s novo prichádzajúcou kótou (Rovnica 5.17).

Hodnota  $n$  značí veľkosť okna a ovplyvňuje celkový vernostný kapitál. Čím väčšia hodnotu nadobúda, tým väčší je vernostný kapitál novo prichádzajúcej kóty (za predpokladu relatívne čistých dát). Avšak vplyv vzdialenejších kót na celkový kapitál sa znižuje, až sa stáva relatívne bezvýznamný. Preto je potrebné voliť vhodnú veľkosť okna. Pri výpočte  $T_{ij}$  je časová vzdialenosť kót v menovateli umocnená ( $\Delta\vartheta_{ij}$ ), čiže so zvyšujúcou vzdialenosťou sa výsledné  $T_{ij}$  blíži nule a neovplyvňuje celkový vernostný kapitál.

### 5.6.1.2 Skokové udalosti

Predchádzajúce výpočty majú jednu podstatnú nevýhodu, ktorú je treba explicitne ošetriť. Prichádzajúcu hodnotu porovnávajú len s predchádzajúcimi hodnotami, lenže problém nastáva v prípade, ak nové – na prvý pohľad zjavne odľahlé hodnoty sú skutočne pravdivé a je to potvrdené hodnotami prichádzajúcimi neskôr (napr. cenový skok). Na odstránenie tohto problému zavedieme špeciálnu alternatívnu hypotézu, ktorá umožní v prípade potvrdenia skoku opraviť chybnú hodnotu vernostného kapitálu a korektne nastaviť on-line štatistiky, aby bola zabezpečená rýchlejšia adaptabilita filtra.

Prvým krokom je lokalizácia možného miesta skoku v skalárnom okne. K tomuto využije predchádzajúci výpočet (Rovnica 5.19), kde pre každé  $j$  testujeme, či čiastkový súčet

$$T'_{i,at\ j} = T_{i0} + \sum_{j'=i-n}^{j-1} C_{j'} T_{ij'}$$

Rovnica 5.20: Výpočet alternatívneho vernostného kapitálu pomocou párového filtrovania a úrovňového filtra

je menší ako kritická prahová hodnota  $T_{crit}$ :

$$T_{crit} = \mu c_{level} T_{i0} - 1$$

Rovnica 5.21: Kritická hodnota akceptovania alternatívneho kapitálu

$\mu$  je malá konštanta, ktorou vykonáme potrebnú normalizáciu. Zároveň testujeme, či  $T_{ij} > 0$ , čo nám indikuje, že sme dosiahli novej stabilnej hodnoty prichádzajúcich kót a teda nejedná sa o kóty s odľahlou hodnotou. Prvá hodnota  $j$  (kumulujeme od najnovších kót k najstarším kótam), na ktorej príde k splneniu uvedených podmienok, označuje miesto v okne, kedy je nová úroveň hodnoty platná. Čiže skutočný skok nastal skorej. Miesto skoku nateraz definujeme ako  $j_{jump} = j$ , avšak je jasné že

skok nastal pred touto hodnotou. Na overenie hypotézy skoku stanovíme alternatívny vernostný kapitál  $T_i''$ . K jeho výpočtu potrebujeme upravený vzťah pre výpočet alternatívneho vernostného kapitálu v mieste skoku

$$T_{i,at\ j_{jump}}'' = T_{crit} - 0,5 + \mu (T_{i,at\ j}' - T_{crit})$$

Rovnica 5.22: Alternatívny párový vernostný kapitál v mieste potenciálneho skoku

Posun o -0,5 je z dôvodu, aby alternatívna hypotéza nebola príliš ľahko akceptovaná, nová úroveň hodnôt musí byť teda podporená viacerými kótami v porovnaní s normálnym výpočtom. Výsledný alternatívny vernostný kapitál vyrátame ako

$$T_i'' = T_{i,at\ j_{jump}}'' + \sum_{j=j_{jump}}^{i-1} C_j'' T_{ij}$$

Rovnica 5.23: Celkový alternatívny vernostný kapitál

kde  $C_j''$  je alternatívna vierohodnosť  $j$ -tej kóty (Rovnica 5.1) vyrátaná zo zmenšeného vernostného kapitálu  $T_j''$ , ktorý vyrátame:

$$T_j'' = \mu T_j$$

Rovnica 5.24: Zmenšený vernostný kapitál  $j$ -tej kóty

Nakoniec rozhodneme, či hodnote výsledného vernostného kapitálu novej kóty  $T_i$  priradíme normálny  $T_i'$  alebo alternatívny (skokový) vernostný kapitál  $T_i''$

$$T_i = \begin{cases} T_i'', & \text{ak } T_i'' > T_i' \text{ a } T_i'' > 0 \\ T_i', & \text{inak} \end{cases}$$

Rovnica 5.25: Výsledný vernostný kapitál novo prichádzajúcej kóty

### 5.6.1.3 Aktualizácia skalárneho okna

Nové kóty ovplyvňujú hodnoty vernostného kapitálu starších kót nachádzajúcich sa v okne. Ak prichádzajúca hodnota sa nachádza na tej istej úrovni ako hodnoty v okne, tak sa ich vernostný kapitál posilňuje. Zásadná zmena hodnôt v okne nastáva v prípade prijatia skokovej hypotézy, kedy musí prísť k výrazným úpravám hodnôt vernostných kapitálov. Pre kóty nachádzajúce sa pred skokom musíme previesť korekciu párového kapitálu

$$T_{corr,ij} = \begin{cases} \mu T_{ij}, & \text{ak } j < j_{jump} \\ T_{ij}, & \text{inak} \end{cases}$$

Rovnica 5.26: Korekcia párového vernostného kapitálu pre kóty pred skokom

Následne kóty nachádzajúce sa za skokom dostávajú znova šancu (vyrátaná hodnota  $T_{j,new}$  nahradí aktuálnu hodnotu  $T_j$ )

$$T_{j,new} = \begin{cases} \mu T_j, & \text{ak } j \geq j_{jump} \text{ a } T_j < 0 \\ T_j, & \text{inak} \end{cases}$$

Rovnica 5.27: Korekcia celkového vernostného kapitálu pre kóty po skoku

Výsledný kapitál všetkých kót  $j$  nachádzajúcich sa v okne je ešte upravený pomocou výsledkov párového filtrovania

$$T_{j,new} = T_j + \sum_{i=j-n}^{j-1} C_i T_{corr,ij}$$

Rovnica 5.28: Úplná korekcia celkového vernostného kapitálu

$C_i$  je vierohodnosť vyrátaná pomocou vernostného kapitálu  $T_i$  (Rovnica 5.1). Hodnotu  $T_j$  opäť nahradíme novou vyrátanou hodnotou  $T_{j,new}$ , čo je výsledný vernostný kapitál aktualizovanej kóty, ktorý sa avšak príchodom nových kót bude ďalej meniť.

V tejto časti sa najvýraznejšie rozlišuje historický a real-time mód filtra:

- *Historický mód*: Dávnejšie zamietnuté kóty (s nízkou vierohodnosťou) nachádzajúce sa v okne môžu byť zmenené na vierohodné. Výstup z filtra bude až po opustení okna.
- *Real-time mód*: Podobne dávnejšie zamietnuté kóty sa môžu stať vierohodnými, avšak výstup filtra je pri ukončení fázy výpočtu, preto na výstupe budú tieto kóty zamietnuté. Napriek tomu je aktualizácia hodnôt v okne potrebná pre správne posúdenie ďalších prichádzajúcich kót.

Aktualizujeme hodnotu životnosti kóty  $Q_j$  všetkých kót nachádzajúcich sa v okne podľa vzťahu

$$Q_{j,new} = Q_j + C_i$$

Rovnica 5.29: Aktualizácia hodnoty životnosti

Hodnotu životnosti novej kóty nastavíme na  $Q_i = 0$ .

Až na konci tejto fázy prichádza samotnému vloženiu novej ( $i$ -tej) kóty do skalárneho okna a veľkosť okna sa zväčšuje o 1. V prípade real-time módu môžeme reportujeme kótu nadradenej vrstve ako konečný výsledok.

#### 5.6.1.4 Výstup hodnôt zo skalárneho okna

Samozrejme okno hodnôt nemôže rásť donekonečna, preto je na konci každého vloženia aplikovaný nasledovný algoritmus, ktorý rieši vyradovanie najstarších kót z okna. Kóty nie sú vyradované pravidelne – v jednom cykle nemusí byť vyradená žiadna alebo môže byť vyradených súčasne viacero. Musíme ale dbať na dodržanie časovej postupnosti. Veľkosť okna ovplyvňujú 3 kritéria (zoraďené od najmenej po najväčšiu dôležitosť):

1. Dostatočný časový interval
2. Dostatočný počet kót
3. Dostatočná celková vierohodnosť skalárnych kót nachádzajúcich sa v okne

Celková vierohodnosť – súčet vierohodností všetkých kót je najdôležitejším parametrom, preto je vo výslednom vzťahu umocnený šiestou mocninou. Kóty posudzujeme od najstaršej, čiže od začiatku okna. Prepúšťacíu podmienku definujeme:

$$(\vartheta_i - \vartheta_{i-n+1}) n^2 \left( \sum_{j=0}^{n-1} C_{i-j} \right)^6 \geq W$$

Rovnica 5.30: Prepúšťacia podmienka skalárneho okna

$\vartheta$  je časový údaj príchodu kóty na trh,  $n$  je veľkosť okna a  $C$  je vierohodnosť ako bolo definované v prechádzajúcich častiach práce. Parameter  $W$  má časovú dimenziu, nakoľko ostatné kritéria sú bezrozmerné. Je závislý na premennej  $v$  (Rovnica 5.17), ktorá čiastočne stanovuje váhu vzdialenejších kót na kóty nové, preto má zmysel zvyšovať hodnotu  $W$  len do tej miery, kedy staršie kóty sú schopné sa výraznejšie premietnuť do výpočtov novej kóty. Jeho hodnotu je potrebné stanoviť experimentálne podľa požadovanej priemernej veľkosti okna (stanovené na  $10^7$ ). Hodnota parametra  $W$  a teda veľkosť okna má veľký vplyv na požadovaný výpočtový výkon, preto je hodnotu potrebné zvoliť dôkladne.

Variabilná veľkosť okna vyplývajúca z uvedeného vzťahu má za následok nasledovné správanie. V prípade, ak do filtra vstupuje prúd čistých dát, tak je veľkosť okna relatívne malá a výpočet rýchly. V prípade vstupu zašumených dát sa okno začne zväčšovať, výpočet spomaľovať, avšak máme k dispozícii väčšie množstvo kót pre porovnanie, preto dokážeme spoľahlivejšie posúdiť vierohodnosť kót.

Ak je prepúšťacia podmienka (Rovnica 5.30) splnená, prichádza k prepusteniu kóty z okna. Pri prepustení príde k odstráneniu kóty z okna a aktualizácii štatistík, navyše ak filter pracuje v historickom režime, tak musíme túto kótu reportovať vyššej vrstve filtra. Prepúšťacíu podmienku opakujeme dovtedy, kým nastane jej nesplnenie.

#### 5.6.1.5 Aktualizácia štatistík

Pri prepúšťaní kóty z okna je jej vernostný kapitál konečný a už sa ďalej nebude meniť. Preto teraz aktualizujeme zberané on-line štatistiky filtra. Na aktualizáciu použijeme len platné kóty, pri ktorých ich vierohodnosť presiahne nastavenú prahovú hodnotu vierohodnosti ( $C_i > C_{crit\_stat}$ ). V tejto časti môžeme odmietnuť len úplne neplatnú kóty, pretože na trhu môže prísť k neočakávaným udalostiam. Filter musí byť schopný sa na ne adaptovať a k tomu je potrebné zahrnúť tieto kóty do jeho štatistík. Použitá prahová hodnota vierohodnosti je preto menšia ako výstupná hodnota vierohodnosti použitá vo vyššej vrstve filtra. Ak je podmienka vierohodnosti splnená, aktualizujeme štatistiky skalárneho okna.

## 5.6.2 Okno pre spracovanie zloženého tikú

Nachádza sa v strede hierarchie nášho filtra. Okno pozostáva z časovo zoradenej postupnosti plných kót. Plná kóta obsahuje bid cenu, ask cenu a im prislúchajúce časové údaje. Unikátnym identifikátorom okna v rámci jedného filtra je symbol (menový pár), ktorý dané okno spracúva. Jeho úlohou je:

- Vytvorenie troch nezávislých skalárnych okien (bid okno, ask okno a okno rozpätia). Skalárne okná sú úplne oddelené, medzi sebou nekomunikujú, všetky výpočty realizujú nezávisle. Jediným spojovacím mostíkom je práve toto okno pre spracovanie zloženého tikú.
- Rozdelenie zložených tikov na skalárne hodnoty a ich distribúcia jednotlivým skalárnym oknám
- Základný vstupný test (Kapitola 5.6.2.1) – testuje, či kótované hodnoty nie sú mimo ich definičný obor. V takom prípade kótu rovno zamietne a nerozosiela ju do skalárnych okien.
- Absolútne hodnoty kót z trhu transformuje na nové hodnoty pre dosiahnutie lepšej distribučnej funkcie (Kapitola 5.6.2.2).
- Zberá kóty reportované skalárnymi oknami, ktoré ich reportujú nezávisle na sebe (rozpätie príde skorej ako bid a pod.), preto musí počkať až bude mať k dispozícii všetky 3 vierohodnosti.
- Po vypočítaní celkovej vierohodnosti na základe všetkých skalárnych okien reportuje plnú kótu do najvyššej vrstvy filtra.

V tejto časti by bolo možné robiť ďalšie filtračné úkony vyžadujúce kompletne kóty. V mojej práci sa filtrovaniu na tejto úrovni nevenujem, ale môže to byť potenciálnym predmetom jej ďalšieho pokračovania.

### 5.6.2.1 Základný vstupný test

Cieľom tohto prvotného testu je vpustiť do filtra len tie hodnoty, ktoré môžu byť za každých okolností platné. Tým myslíme rôzne extrémne výkyvy a pády trhu. Odfiltrujeme len tie hodnoty, ktoré nedávajú zmysel – sú mimo definičný obor.

- Cena – u ceny platí, že nikdy nemôže byť záporná  $p \geq 0$
- Časová os – tiky musia prichádzať v poradi v akom boli poslané na trh, v opačnom prípade tiky rovno zahadzujeme

### 5.6.2.2 Transformácia absolútnych hodnôt

Filtrované premenné matematicky transformujeme z nasledovných dôvodov:

1. Symetrickejšia distribučná funkcia hodnôt. Matematické operácie vykonávané v skalárnom okne očakávajú striktne symetrickú distribúciu hodnôt a ich zmien. Transformáciou pôvodných hodnôt dosiahneme symetrickejšie rozloženie hodnôt. Využívame logaritmické transformácie.
2. Transformovaná hodnota nesmie byť závislá na mene (rozdiel medzi obvyklými hodnotami USD a JPY). Relatívne zmeny hodnoty sú potom vzájomne ľahšie porovnateľné. Požadovanú vlastnosť opäť realizuje logaritmická transformácia.

Transformácia musí byť navrhnutá podľa základného vstupného testu, kedy definičný obor transformácie musí byť podmnožinou oboru hodnôt sledovanej premennej. Zohľadníme základný vstupný test a teda nevyhovujúce kóty už do transformačného procesu nevstupujú.

Skalárne bid a ask ceny transformujeme podľa vzťahu:

$$x_{tran} = \log(x)$$

Rovnica 5.31: Logaritmická transformácia hodnoty bid a ask ceny

Pre hodnotu rozpätia volíme transformáciu:

$$x_{spread} = \log(x_{ask}) - \log(x_{bid})$$

Rovnica 5.32: Logaritmická transformácia hodnoty rozpätia

Pretože v určitých extrémnych situáciách môže rozpätia nadobúdať aj záporných hodnôt. Tento jav je známy pod pojmom *negatívne rozpätie (negative spread)*.

## 5.6.3 Univariantný filter

Sa nachádza na najvyššej úrovni hierarchie filtra. Je to zároveň miesto kde sú vkladane tiky prichádzajúce z trhu (prípadne iného dátového zdroja). Ako najvyššie úroveň má za úlohu spravovať nižšiu vrstvu, zabezpečuje konfiguráciu filtra, vstup a výstup hodnôt. Filter môže súčasne spracovávať viacero symbolov (menových párov), čo zabezpečíme vytvorením samostatného okna pre filtráciu zloženého tiky pre každý symbol. Avšak ako z názvu vrstvy vyplýva, tieto okná budú filtrovať nezávisle na sebe.

Medzi hlavné úlohy univariantného filtra patrí:

- Správa konfigurácie filtra
- Vytvorenie okien pre filtráciu zloženého tiky pre každý symbol
- Delenie prúdu prichádzajúcich tikov podľa symbolu a ich rozosielanie príslušným oknám strednej vrstvy
- Inicializácia filtra počiatočnou sériou tikov pre jeho adaptáciu pre aktuálny trh
- Formátovanie výsledkov do výstupného formátu
- Distribúcia výsledkov užívateľovi (zápis do súboru)

### 5.6.3.1 Výstup filtra

Výsledok filtrovania je reportovaný ako prúd tikov, ktorý navyše každému tiku priradzuje výsledné hodnoty reportovaný koncovému užívateľovi:

1. Výslednú vierohodnosť tikú
2. Čiastkové vierohodnosti (bid, ask a rozpätia)
3. Platnosť kóty (platná alebo neplatná)

Užívateľ si môže množstvo výsledkov v konfigurácii zredukovať v konfigurácii. Najzákladnejším výstupom je výstup iba platných kót. Rozšírené výstupné módy sú určené pre ladenie filtra a experimenty s ním.

Výslednú vierohodnosť tikú  $C$  určíme na základe jeho čiastkových vierohodností

$$C = \min (C_{bid}, C_{ask}, C_{spread})$$

5.33: Výsledná vierohodnosť tikú

a rozhodneme o celkovej platnosti  $V$  tikú

$$V(C) = \begin{cases} \textit{platný}, & C > C_{crit} \\ \textit{neplatný}, & C \leq C_{crit} \end{cases}$$

5.34: Konečná platnosť tikú

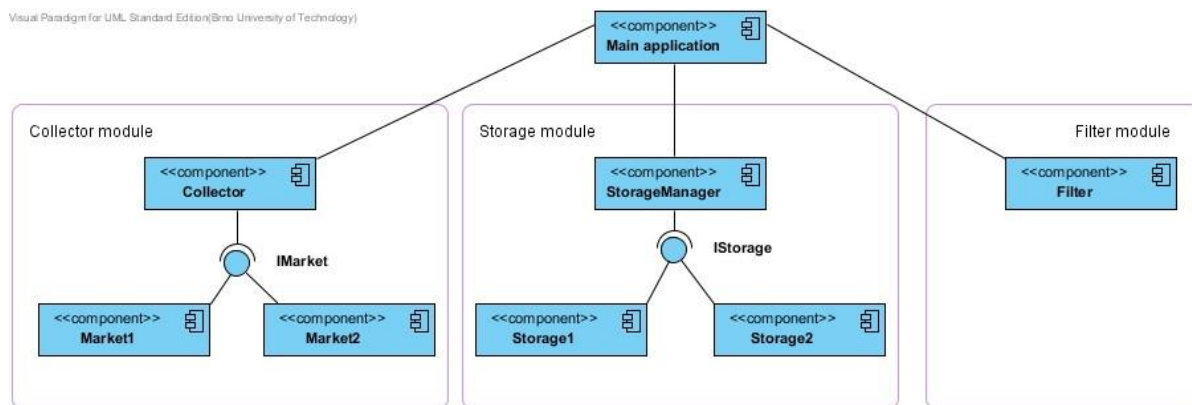
kde  $C_{crit}$  je nastaviteľná prahová hodnota vierohodnosti na intervale  $< 0; 1 >$ . Obvykle volíme hodnoty v intervale  $< 0; 0,5 >$ .

## 6 Implementácia

V tejto kapitole je obsiahnutý výber implementačného prostredia a popis implementácie jednotlivých častí práce. Výsledným produktom práce sú 3 knižnice, ktoré sú ovládané pomocou centrálneho grafického užívateľského rozhrania:

1. Knižnica zberu dát (Collector library)
2. Knižnica uloženia dát (StorageManager library)
3. Filtračná knižnica (Filter library)

Jednotlivé moduly sú na sebe absolútne nezávislé, preto ich je možné zakomponovať do ďalších aplikácií. Pre demonštráciu boli implementované knižnice napojené na grafické užívateľské rozhranie. Vzájomné prepojenie jednotlivých modulov aplikácie znázorňuje Obrázok 6.1.



Obrázok 6.1: Diagram komponent modulárnej architektúry výslednej aplikácie

### 6.1 Implementačné prostredie

V tejto podkapitole sa venujem k popisu použitých implementačných prostredí. Výsledná aplikácia je postavená na Microsoft .NET Framework, implementovaná v jazyku C#. Ako úložisko dát bol vybraný databázový server Microsoft SQL Server. Pre komunikáciu s trhom pomocou protokolu FIX bola použitá otvorená knižnica VersaFix. Užívateľské prostredie bolo realizované pomocou štandardného Windows Forms s open-source nadstavbou DockPanel.

#### 6.1.1 Technológia Microsoft .NET Framework

Použitý .NET Framework je firemnou platformou spoločnosti Microsoft určenou pre vývoj širokej škály aplikácií, od jednoduchších desktopových, cez rozsiahle podnikové aplikácie až po špecializované aplikácie pre rôzne mobilné zariadenia. Informácie boli čerpané prevažne z [8]. Framework pozostáva z 3 hlavných častí:



- Common language run-time (CLR) – poskytuje abstrakciu služieb operačného systému (pre dynamické jazyky je k dispozícii Dynamic language run-time (DLR))
- Base class libraries – naprogramované kódy pre často používané funkcie nižšej úrovne na nižšej úrovni programovania
- Vývojárske frameworky a technológie – znovu použiteľné a užívateľsky prispôsobiteľné komponenty pre budovanie rozsiahlych systémov

Framework poskytuje komplexné a ucelené vývojárske prostredie s množstvom pripravených najpoužívanejších funkcií. Jednou z veľkých devíz je nezávislosť na použitom programovacom jazyku, podporovaných je hneď viacero. Medzi statické jazyky patria C# a Visual Basic, z podporovaných dynamických spomeniem F#, IronRuby a IronPython.

Pre mobilné zariadenia je k dispozícii odľahčená verzia .NET Framework Compact a .NET Framework Micro pre vstavané zariadenia.

Obsahuje nástroje pre vývoj moderných užívateľských rozhraní. Windows Forms je klasické užívateľské rozhranie pre desktopové aplikácie. Windows Presentation Foundation (WPF) je určený pre dizajn moderných desktop grafických užívateľských rozhraní, na popis rozhrania používa jazyk XAML, čo umožňuje výrazne lepšie oddeliť grafickú časť aplikácie od jej aplikačnej logiky. Grafik a programátor môžu pracovať nezávisle, spolu riešia len vzájomnú integráciu jednotlivých častí. Obdobou WPF je technológia Silverlight, ktorý je určený pre moderné interaktívne internetové aplikácie. Klasické internetové stránky pokrýva technológia ASP.NET.

Tvorbu aplikácií založených na službách obstaráva vrstva Windows Communication Foundation (WCF), ktorá zapuzdruje potrebné prostriedky pre vytvorenie rozsiahlych a bezpečných architektúr.

Dátové entity sprístupňuje technológia ADO.NET, ktorá poskytuje prostriedky pre vytvorenie rozsiahlych aplikácií pracujúcich s dátami. Rovnako umožňuje mapovanie dátových štruktúr priamo na jednotlivé triedy a výrazne tým uľahčuje prácu vývojárovi. Language Integrated Query (LINQ) je rozšírením platformy .NET o možnosť priameho výberu a ukladania dát pomocou jednoduchých príkazov a uľahčuje tak celkové prepojenie dátovej vrstvy s aplikačnou. LINQ je použiteľný pre veľké množstvo kolekcí (databázy, XML, zoznamy ... ).

Rozsiahle možnosti platformy sú integrované do moderného vývojárskeho prostredia Visual Studio, ktoré umožňuje efektívne využívať poskytnuté možnosti a výrazne tak zefektívňuje vývoj projektov.

Ako implementačný jazyk som zvolil C#, pretože sa jedná o typovo bezpečný objektovo orientovaný jazyk, ktorý je na riešenie tejto úlohy vhodný. Aj keď efektivita je dôležitá, aplikácia nie je jej kritickým parametrom, preto nie je nutné pre kritické časti používať efektívnejší kód v jazyku C++ alebo assembler ako u iných HFT systémov.

## 6.1.2 Microsoft SQL Server

Microsoft SQL Server je relačným databázovým serverom spoločnosti Microsoft, ktorý ponúka ucelené komplexné portfólio služieb. Je k dispozícii široká škála verzií pre aplikácie rôznej zložitosti. Portfólio verzií ponúka odľahčenú verziu pre mobilné a nenáročné aplikácie (SQL Server Compact), cez voľne dostupnú verziu s obmedzeným počtom ponúkaných služieb (SQL Server Express) až po komplexnejšie produkty slúžiace na ukladanie veľkého objemu dát, nad ktorými umožňuje vykonávať rôzne reporty, analytické operácie, dolovanie dát a s tým súvisiace nástroje pre Business Intelligence (BI). Z pokročilejších funkcií servera budeme využívať službu replikácie dát, ktorá dokáže synchronizovať databázy bežiacie na rôznych serveroch.

Požiadavky server prijíma v jazyku ANSI SQL a T-SQL (Transact SQL), čo je proprietárnym rozšírením pôvodného jazyka SQL spoločnosťou Microsoft o nové prvky. Pre správu servera slúži grafická konzola SQL Server Management Studio. K Microsoft SQL Servu je možné pristupovať z rôznych technológií, avšak najjednoduchšiu integráciu zabezpečuje spojenie s .NET aplikáciou.

Microsoft SQL Server som zvolil z dôvodu vyhovujúcich vlastností potrebám budovanej aplikácie, jeho rozšírenosti a predchádzajúcich pozitívnych skúsenosti. Pre fungovanie aplikácie je potrebná minimálne verzia Express, ak chceme využívať synchronizácie medzi databázami potrebujeme vyššiu platenú verziu. V rámci študijného programu MSDN som použil vývojársku verziu Developer, určenú len pre vývoj [9].

## 6.1.3 VersaFix

VersaFix<sup>12</sup> je open-source knižnica vyvíjaná špeciálne pre použitie na platforme .NET. Knižnica slúži pre vývoj vlastných aplikácií komunikujúcich s trhom pomocou protokolu FIX. VersaFIX je nadstavbou nad TCP spojením, ktoré je využívané pre komunikáciu s trhom. Tento systém nám umožňuje efektívne posielat' a prijímať správy, vytvára spojenie s trhom a pravidelne ho kontroluje. Systém nám umožňuje sa plne zamerať na aplikačnú logiku aplikácie.

Bohužiaľ už počas vývoja diplomovej práce prišlo k veľkej zmene v smerovaní projektu VersaFix, ktorý sa stal komerčne zameraný<sup>13</sup>. Spoločnosť Assimilate Technology ponúka rozsiahle graficky prívetivé vývojárske nástroje, avšak tieto už sú spoplatnené. Podľa dostupných informácií nič nenasvedčuje tomu, že by vývoj pôvodného open-source systému pokračoval. Avšak jeho funkcionálnosť je aj v poslednej dostupnej beta verzii pre potreby práce postačujúca.

---

<sup>12</sup> Open-source verzia: <http://sourceforge.net/projects/versafix/>

<sup>13</sup> Komerčná verzia: <http://www.assimilate.com/>

## 6.2 Collector - knižnica zberu dát

Modul zberu tvorí samostatnú knižnicu s názvom Collector. Knižnica sama o sebe nepodporuje žiadny trh, ale definuje rozhranie, pomocou ktorého je možné implementovať ďalšie trhy ako zásuvné moduly. Knižnica je nezávislá na použitom komunikačnom protokole. Použitý protokol je súčasťou implementácie zásuvného modulu, preto je možné realizovať súčasne zber dát z viacerých trhov fungujúcich pod rôznymi komunikačnými protokolmi. V rámci implementácie mojej práce som si vybral trh spoločnosti HotSpot využívajúci komunikačného protokolu FIX.

### 6.2.1 FIX protokol

Financial Information Exchange (FIX) protokol je štandardizovaný formát pre elektronickú komunikáciu v oblasti finančných trhov. Bol vyvinutý skupinou bánk, brokerských spoločností, investorov a IT spoločnosťami z rôznych častí sveta. Potrebu obchodníkov bolo používanie jedného komunikačného štandardu všetkými spoločnosťami pre všetky bežné finančné operácie. FIX je najpoužívanejším protokolom v oblasti elektronického obchodu, z implementačného pohľadu predstavuje časovo rýchle a finančne nenáročné riešenie. Takisto existuje veľké množstvo hotových produktov, preto časť obchodníkov sa môže využívať už ponúkané produkty bez nutnosti zaoberať sa FIX protokolom. Zdrojom pre informácie o protokole FIX bola oficiálna stránka projektu [10].

#### 6.2.1.1 Verzie protokolu FIX

Postupný vývojom protokolu vznikali jednotlivé verzie protokolu, ktoré sa odlišovali funkčnosťou<sup>14</sup>. Verzie nie sú 100% spätne kompatibilné, preto je vždy potrebné pracovať s verziou, ktorú vyžaduje na komunikáciu konkrétny elektronický trh. Avšak jednotlivé trhy môžu poskytovať komunikáciu vo viacerých verziách súčasne a klient si len vyberie verziu, ktorá mu viac vyhovuje.

Dátum vydania	Január 1996	Apríl 1998	Marec 2000	August 2001	August 2003	Október 2006	Marec 2008	Apríl 2009
Verzia protokolu	4.0	4.1	4.2	4.3	4.4	5.0	5.0 SP1	5.0 SP2

Tabuľka 6.1: Prehľad verzií protokolu FIX

V práci implementujem komunikáciu s trhom HotSpot, ktorý používa protokol verzie 4.2, preto ak nebude uvedené inak, všetky ďalšie informácie sa budú týkať uvedeného protokolu.

<sup>14</sup> Kompletný prehľad obsiahnutej funkcionality jednotlivých verzií protokolu FIX je dispozícii na <http://www.fixprotocol.org/functionality>

### 6.2.1.2 Formát správy

Správy sú posielané pomocou soketov nad TCP sieťovou vrstvou. Protokol FIX je teda platformovo nezávislý. Jednotlivé polia v správe oddeľujeme ASCII znakom s kódom 01 (<start of header>), ktorý budeme ďalej nazývať oddeľovač. Výsledná správa sa skladá z 3 častí:

- Header (hlavička správy)
- Body (telo správy)
- Trailer (päta správy)

Obsah správy je textový a jednotlivé informácie sú ohraničené oddeľovačom. Informácia pozostáva z dvojice údajov: *značka* (tag) a *hodnota* (value). Napríklad v údaji 35=V, číslo 35 reprezentuje typ správy (MsgType) a V jeho hodnotu, ktorou je požiadavka na príjem dát z trhu (MarketDataRequest). Význam jednotlivých polí a ich hodnôt, ktoré môžu nadobúdať je dostupný vo voľne prístupných FIX slovníkoch (angl. FIX dictionary). Ďalšou výhodou protokolu je jeho rozšíriteľnosť. V prípade ak vývojárovi nepostačuje štandardizovaný slovník, môže si dodefinovať vlastné páry údajov. Rovnako im je možné nastaviť atribúty ako u štandardizovaných údajoch.

*Hlavička správy* pozostáva z polí, ktoré uvádza Tabuľka 6.2. Túto hlavičku musí obsahovať každá FIX správa a všetky uvedené položky musí vytvorená správa obsahovať. Súčasťou definície sú aj nepovinné polia, ktoré môže byť špecificky využit' určitou platformou. Napriek tomu, že je pole nepovinné v definícii FIX protokolu, môže byť povinne vyžadované pre konkrétny trh.

Značka	Názov	Komentár
8	BeginString	Verzia FIX protokolu
9	BodyLength	Dĺžka tela správy
		Typ správy:
		0 - HeartBeat
		A - Logon
35	MsgType	V - MarketData request
		W - Snapshot/Full refresh
		X - Incremental refresh
34	MsgSeqNum	Postupne sa zväčšujúca hodnota (nová správa = predchádzajúca správa + 1)
49	SenderCompID	ID spoločnosti odosielateľa
56	TargetCompID	ID cieľovej spoločnosti
52	SendingTime	GMT čas odoslania

Tabuľka 6.2: Hlavička správy podľa protokolu FIX 4.2

*Telo správy* je špecifické pre každú hodnotu MsgType. Špecifikácia požadovaných polí tela a ich hodnôt je opäť súčasťou slovníka. Podrobnejšie sa telu správy pre použité správy pri implementácii diplomovej práce venujem v nasledujúcej kapitole.

*Päta správy* obsahuje kontrolný súčet a slúži na overenie integrity prichádzajúcich dát na strane prijímateľa.

Značka	Názov	Hodnota	Komentár
10	Checksum		Súčet bajtov všetkých polí správy

Tabuľka 6.3: Päta správy podľa protokolu FIX 4.2

### 6.2.1.3 Typy správ

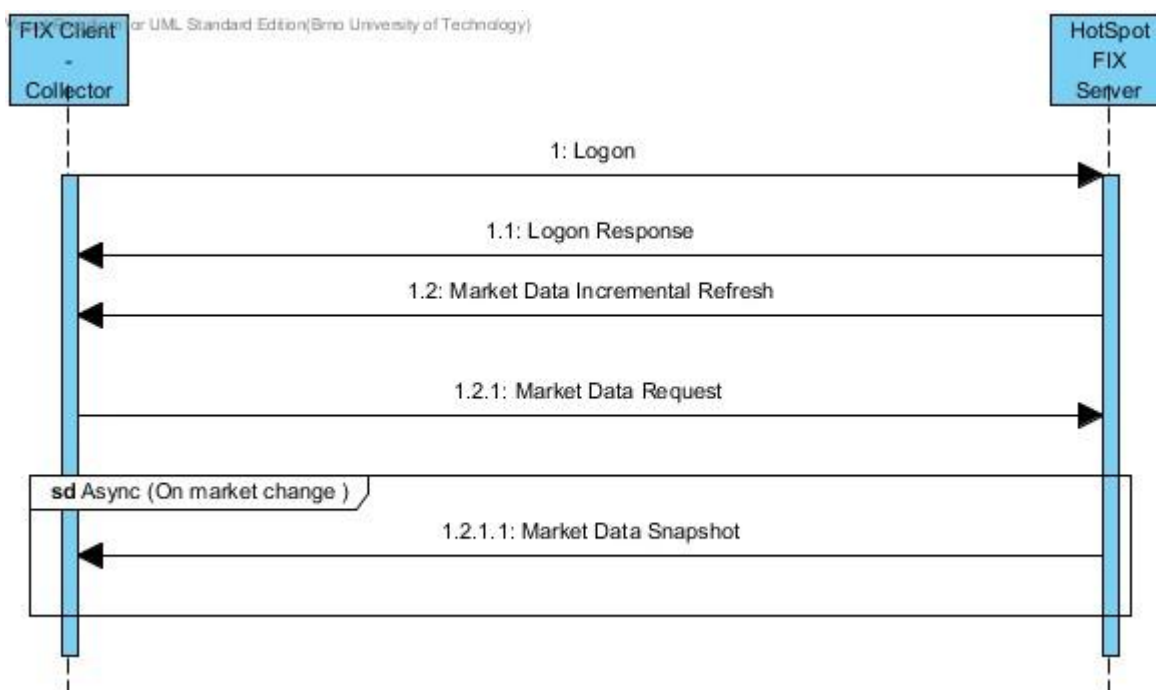
Podľa významu správy delíme na *administratívne* a *aplikačné*. Obidva druhy dodržiavajú uvedený formát, avšak ich rozčlenenie je dôležité z hľadiska lepších možností spravovania obchodného systému.

*Administratívne*, niekedy nazývané aj relačné, správy súvisia so správou spojenia medzi klientom a serverom, a zabezpečujú autentizáciu. Rovnako sem patria rôzne chybové správy súvisiace so spojením.

*Aplikačné* správy súvisia priamo s obchodovaním. Jedná sa o správy obsahujúce informácie o situácii na trhu (aktuálne ceny), realizované obchody, zadávanie ponuky na trh alebo reakcia na tržnú ponuku.

## 6.2.2 Hotspot FIX BookFeed API

Je aplikačné rozhranie spoločnosti HotSpot fungujúce pomocou špecifikácie FIX protokolu. Je definované v on-line dokumente [11]. V tejto kapitole bude vysvetlená komunikácia s trhom, sled správ potrebných na prihlásenie sa k trhu a odber obchodných dát. Ďalej bude popísaný formát prijatých správ a ich spracovanie. Názočne komunikáciu s trhom znázorňuje nasledovný sekvenčný diagram (pre jednoduchosť neobsahuje možné chybové správy)



Obrázok 6.2: Sekvenčný diagram komunikácie klientskej aplikácie so serverom obchodnej spoločnosti Hotspot

### 6.2.2.1 Prihlásenie klienta

Samotné spojenie realizované nad TCP vrstvou nám zabezpečuje použitý VersaFix. K správnomu pripojeniu je potrebné nastaviť prístupové údaje. Po úspešnej inicializácii sieťového spojenia musíme odoslať žiadosť o prihlásenie, ktorú realizujeme pomocou správy *Logon*. Jednotlivé položky prihlasovacej správy znázorňuje Tabuľka 6.4.

Značka	Názov	Dátový typ	Komentár
35	MsgType	Char	'A' - Logon
553	Username	String	Prihlasovacie meno
554	Password	String	Heslo
98	EncryptMethod	Integer	0
108	HeartBtInt	Integer	30 – interval 30 sekúnd

Tabuľka 6.4: Prihlasovacia správa - od klienta (Logon)

Ako odpoveď od servera príde správa s hodnotou *MsgType=A – Logon Response*, čo značí úspešné prihlásenie k serveru. V prípade neúspechu prichádza zamietavá správa *MsgType=3* s podrobnejším popisom chyby obsiahnutom v tele správy.

Značka	Názov	Dátový typ	Komentár
35	MsgType	Char	'A' - Logon
98	EncryptMethod	Integer	0
108	HeartBtInt	Integer	30 – interval 30 sekúnd

Tabuľka 6.5: Prihlasovacia správa - odpoveď zo servera (Logon Response)

### 6.2.2.2 Priebežná kontrola spojenia

Klient v pravidelnom intervale posíla správu *Heartbeat* (*MsgType=0*), na ktorú server odpovedá testovacím tikom *TestRequest* (*MsgType=1*). Toto nám umožňuje identifikovať prípadné výpadky spojenia medzi klientom a serverom.

### 6.2.2.3 Dostupné symboly na trhu

Po úspešnom prihlásení trh automaticky posíla správu obsahujúcu všetky dostupné symboly. Táto správa predchádza všetkým správam s *MsgType W*.

Tag	Názov	Dátový typ	Komentár
35	MsgType	Char	'X' – Market Data Incremental Refresh
268	NoMDEntries	Integer	Počet symbolov
55	Symbol	String	Táto skupina sa opakuje pre každý symbol
279	MDUpdateAction	Char	
5071	TickBandNoDecPlaces	Integer	

Tabuľka 6.6: Informácie o dostupných symboloch – automaticky poslaná správa od servera

#### 6.2.2.4 Žiadosť o obchodné dáta

K odberu dát sa prihlasujeme správou *Market Data Request*, v ktorej sú obsiahnuté požadované symboly. Pre každý symbol ako odpoveď zo strany servera príde správa *Market Data Snapshot*.

Požadovaný počet symbolov určuje značka *NoRelatedSym*. Pre každý symbol je v tele správy obsiahnutá skupina údajov (Tabuľka 6.7). *Symbol* (značka 55) je požadovaný menový pár zapísaný vo formáte CCY1/CCY2, kde jednotlivé políčka zodpovedajú ISO formátu zápisu jednotlivých mien (napr.: EUR/USD). *MDReqIDt* (značka 262) je unikátnym identifikátorom správy, unikátnosť je zabezpečovaná na strane klienta. *SubscriptionReqType* (značka 263) nastavíme na hodnotu Snapshot + Update (hodnota 1), teda príjem celej knihy vrátane neskorších aktualizácií. *Market depth* (značka 264) vyjadruje počet odoberaných úrovní z knihy objednávok – 0 pre celú knihu, 1..n podľa požadovaného počtu úrovní. V našej aplikácii zberáme len vrch knihy objednávok (top of book), teda použijeme hodnotu 1. *MDEntryType* je v implementácii HotSpot ignorované, preto hodnotu *NoMDEntryTypes* nastavíme na 0. Detailnejší popis poskytne slovník k protokolu.

Tag	Názov	Dátový typ	Komentár
35	MsgType	Char	'V' – Market Data Request
146	NoRelatedSym	Integer	Počet symbolov
55	Symbol	String	
262	MDReqIDt	String	
263	SubscriptionReqType	Char	Táto skupina sa opakuje
264	MarketDepth	Integer	pre každý symbol
267	NoMDEntryTypes	Integer	
269	MDEntryType	Char	

Tabuľka 6.7: Žiadosť o dáta - od klienta (Market Data Request)

V prípade trhu HotSpot je po pripojení automaticky spustený odber všetkých menových párov. Pri zmene je potrebné predchádzajúci odber zrušiť, *SubscriptionReqType* (značka 263) nastavíme na hodnotu 2, ktorá zruší predchádzajúci odber. Následne pošleme správu obsahujúcu symboly nového odberu.

#### 6.2.2.5 Príjem obchodných dát

Správa *Market Data Snapshot* obsahuje informácie o situácii na trhu vo forme knihy objednávok a správa je zasielaná asynchrónne klientovi v prípade jej zmeny. Z trhu prichádzajú len tie menové páry, ktorých odber máme prihlásený. Tieto správy môžeme kategorizovať do 2 skupín:

1. *Aktualizácia knihy objednávok* – obsahuje informácie o aktuálnych bid alebo ask ponukách daného symbolu (*MDEntryType* - 0 = bid, 1 = ask), cene (*MDEntryPx*), požadovanom množstve (*MDEntrySize*) a počte ponúk (*NumberOfOrders*). Formát správy znázorňuje Tabuľka 6.8.

Značka	Názov	Dátový typ	Komentár
35	MsgType	Char	'W' – Market Book Update
55	Symbol	String	Menový pár
268	NoMDEntries	Integer	Počet cien v správe
269	MDEntryType	Char	Táto skupina sa opakuje pre každú položku knihy objednávok
270	MDEntryPx	Double	
271	MDEntrySize	Double	
346	NumberOfOrders	Integer	

Tabuľka 6.8: Aktualizácia knihy objednávok - asynchrónne od servera

2. *Informácia o realizovanom obchode* – obsahuje informáciu o uskutočnenom obchode, ktorý identifikujeme pomocou *MDEntryType* s hodnotou 2 (*Trade*). Obchodované množstvo *MSEntrySize* je nastavené na 0. *TickDirection* udáva smer realizovaného obchodu, hodnota 0 znamená nákup a hodnota 2 predaj daného menového páru.

Značka	Názov	Dátový typ	Komentár
35	MsgType	Char	'W' – Market Book Update
55	Symbol	String	Menový pár
268	NoMDEntries	Integer	Počet cien v správe
269	MDEntryType	Char	Táto skupina sa opakuje pre každú položku knihy objednávok
270	MDEntryPx	Double	
271	MDEntrySize	Double	
272	MDEntryDate	UTCDate	
273	MDEntryTime	UTCTimeOnly	
274	TickDirection	Char	

Tabuľka 6.9: Informácia o realizovanom obchode - asynchrónne od servera

## 6.3 Storage - knižnica pre uloženie dát

Uloženie zberaných dát vo výslednej aplikácii je tvorené samostatnou knižnicou s názvom *Storage*. Rovnako ako v prípade knižnice pre zber dát, samotná knižnica neumožňuje uloženie dát. Jednotlivé úložiská sú riešené formou zásuvných modulov, čo umožňuje implementovať veľmi rozličné a špecifické úložiská. V mojej práci som implementoval úložisko pre MS SQL Server a úložisko pre zápis dát do súboru (FileWriter). Konfigurácia je realizovaná prostredníctvom XML súborov.

Zápis dát do úložísk prebieha na základe udalosti vyvolanej zberovým modulom. Udalosť na trhu je prijatá centrálnym modulom, ktorý o prijatej udalosti informuje modul úložiska *Storage*. Ten následne pre všetky vytvorené inštancie úložísk zavolá metódu aplikujúcu uloženie tiketu.

### 6.3.1 Redundantný zápis

Zápis teda prebieha paralelne do viacerých úložísk. Dôvodom tohto riešenia je požiadavka na kontinuálny dlhotrvajúci zápis dát a ochrana proti možným výpadkom. Redundantný zápis chráni pred výpadkom niektorého z úložných zariadení, vtedy budú dáta uložené na druhom funkčnom zariadení. Následne ich je možné manuálne preniesť do úložiska kde chýbajú. V prípade MS SQL



Servera sa v práci zaoberám aj automatickou synchronizáciou chýbajúcich dát (Kapitola 6.3.2). Zápis je možný realizovať na ľubovoľné množstvo zariadení.

### 6.3.2 Modul - zápis do súboru

Zásuvný modul *FileWriter* realizuje zápis zberaných dát do CSV súboru s oddeľovačom. Súbor sa ukladajú do adresára nastaveného v konfiguračnom súbore. Názov súboru je v tvare *dátum\_čas.csv*, kde dátum a čas sú hodnoty spustenia zápisu. Výstupný formát kopíruje štruktúru tabuľky *Tick* v databáze (Obrázok 4.1) a pozostáva z 10 stĺpcov:

- Symbol
- BidPrice
- AskPrice
- BidQty
- AskQty
- BidOrders
- AskOrders
- BidTimeStamp
- AskTimeStamp
- StreamID

### 6.3.3 Modul – uloženie do databázy

Zásuvný modul Microsoft SQL Server ukladá zbierané dáta do databázy. Databáza už musí obsahovať vytvorené tabuľky vrátane hodnotami naplnených pomocných tabuliek. Štruktúra databázy odpovedá ER diagramu prezentovanému v návrhu aplikácie (Obrázok 4.1). Konfigurácia pozostáva z pripojovacieho reťazca (angl. connection string) pre MS SQL Server<sup>15</sup>. Pri vkladaní dát využívame z výkonnostných dôvodov uloženú procedúru.

### 6.3.4 Synchronizácia databáz

Informácie k synchronizácii databáz boli čerpané z oficiálnej dokumentácie spoločnosti Microsoft [12]. V prípade výpadku dát v niektorej z databáz je možné dáta manuálne doplniť. Microsoft SQL Server však poskytuje špecializované replikačné nástroje. Replikácia je množina technológií slúžiacich na kopírovanie, distribúciu dát medzi databázami (servermi). Pri správnej konfigurácii zabezpečuje konzistentný stav synchronizovaných databáz. Pomocou replikácie je možné distribuovať dáta do databáz s rôznymi technológiami pripojenia k sieti. Úlohy serverov pri replikácii si môžeme definovať nasledovne:

1. *Vydavateľ* (angl. *publisher*) – server poskytujúci dáta pre replikáciu. Jeden vydavateľ môže obsahovať viacero *publikácií*. Publikácia je definovaná jednotka replikácie s vlastnými pravidlami a logikou.

---

<sup>15</sup> Zoznam pripojovacích reťazcov je dispozícii na <http://www.connectionstrings.com>

2. *Príjemca* (angl. *subscriber*) – server, ktorý dáta naopak prijíma a aktualizuje svoje tabuľky. Príjemca môže prijímať dáta od viacerých vydavateľov. V niektorých typoch replikácie príjemca môže dáta posielat' späť vydavateľovi alebo posielat' ďalším uzlom.
3. *Distribútor* (angl. *distributor*) – uchováva špecifické dáta súvisiace s jedným alebo viacerými vydavateľmi. Každý vydavateľ má práve jedného distribútora. Distribútor ukladá informácie o stave replikácie, metadáta o publikácii a vytvára frontu pre dáta posielané vydavateľom k prijímateľovi. Vo väčšine prípadov je distribútor priamo súčasťou vydavateľa – hovoríme o *lokálnom* distribútorovi. V prípade, že distribútor je umiestnený na inom serveri – hovoríme o *vzdialenom* distribútorovi.

Replikácie podľa spôsobu realizácie delíme do 3 základných skupín.

*Transakčná replikácia* sa obyčajne používa medzi servermi v situáciách vyžadovanej vysokej priepustnosti, zvyšovania dostupnosti, práca s dátovými skladmi, integrácia dát z viacerých zdrojov, integrácia heterogénnych dát a vyrovňovania zátáže.

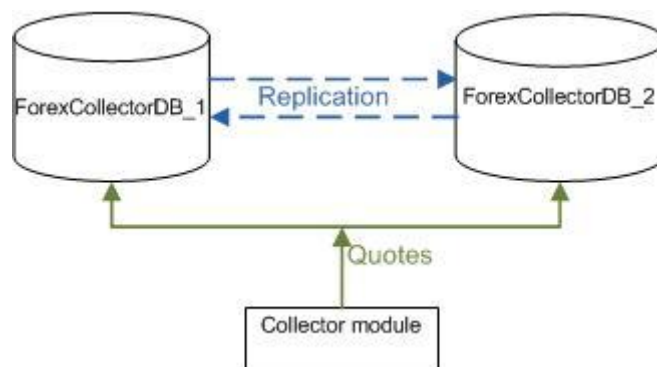
*Zlučovacia* (angl. *merge*) *replikácia* je určená pre mobilné aplikácie alebo aplikácie, ktoré nie sú trvalo pripojené k sieti a môžu tak vznikať dátové konflikty. Typickým využitím je výmena dát s mobilnými užívateľmi.

*Snímková* (angl. *snapshot*) *replikácia* slúži na počiatočnú synchronizáciu veľkého množstva dát ďalej synchronizovaných pomocou transakčnej alebo zlučovacej replikácie. Ďalším využitím je explicitné spustenie aktualizácie dát.

Tieto základné replikácie obsahujú ešte ďalšie varianty. Voľba konkrétnej varianty závisí od topológie siete a usporiadania vzťahov medzi jednotlivými databázovými servermi.

#### **6.3.4.1 Obojsmerná transakčná replikácia**

Pre implementáciu som si zvolil obojsmernú transakčnú replikáciu medzi 2 servermi – jedná sa o spojenie typu peer-to-peer [13]. Do obidvoch databáz sú zberovým modulom ukladané tiky s unikátnym 16 bitovým identifikátorom. Pojem obojsmerná znamená, že chýbajúce kóty môžu byť nahraditeľné v prípade výpadku hociktorej z databáz. K ich výpadku pochopiteľne nesmie prísť súčasne, potom by už dáta neboli obnoviteľné. Implementovanú situáciu znázorňuje Obrázok 6.3. Čiže ak nastane zmena v databáze ForexCollectorDB\_1, tak jeho vydavateľ informuje prijímateľa ForexCollectorDB\_2 o vzniknutej zmene v databáze. Rovnako to platí v zrkadlovom prípade. V zavedenej terminológii to znamená, že obidva serveri budú súčasne vydavateľom a prijímateľom tých istých dát. Vydavateľ posielá zmeny vykonané v jeho databáze, avšak nesmie poslať zmeny vykonané párovým vydavateľom, pretože by vznikla nekonečná slučka.



Obrázok 6.3: Schéma obojsmernej replikácie

Nakoľko do obidvoch databáz modul zberu zapisuje paralelne, pri zápise vznikajú dva konflikty:

1. Zápis zo zberového modulu prebehne replikáciu – je potrebné testovať uloženou procedúrou replikácie existenciu danej kóty v tabuľke pomocou unikátneho identifikátora.
2. Replikácia prebehne zberový modul - je potrebné testovať uloženou procedúrou replikácie existenciu danej kóty v tabuľke pomocou unikátneho identifikátora.

Z uvedeného vyplýva, že je vhodné použiť univerzálnu procedúru aj pre replikáciu aj pre vkladanie dát do databázy pomocou zberového modulu. Konfiguračný súbor replikácie je k dispozícii na priloženom CD.

## 6.4 Filter

Knižnica filtra je nazvaná Filter a ktorá implementuje matematicko-štatistický model popísaný v Kapitole 5. Filtračný proces je nezávislý na zdroji dát. V práci som implementoval filtráciu z dát uložených v databáze, avšak knižnicu je možné napojiť na ľubovoľný zdroj.

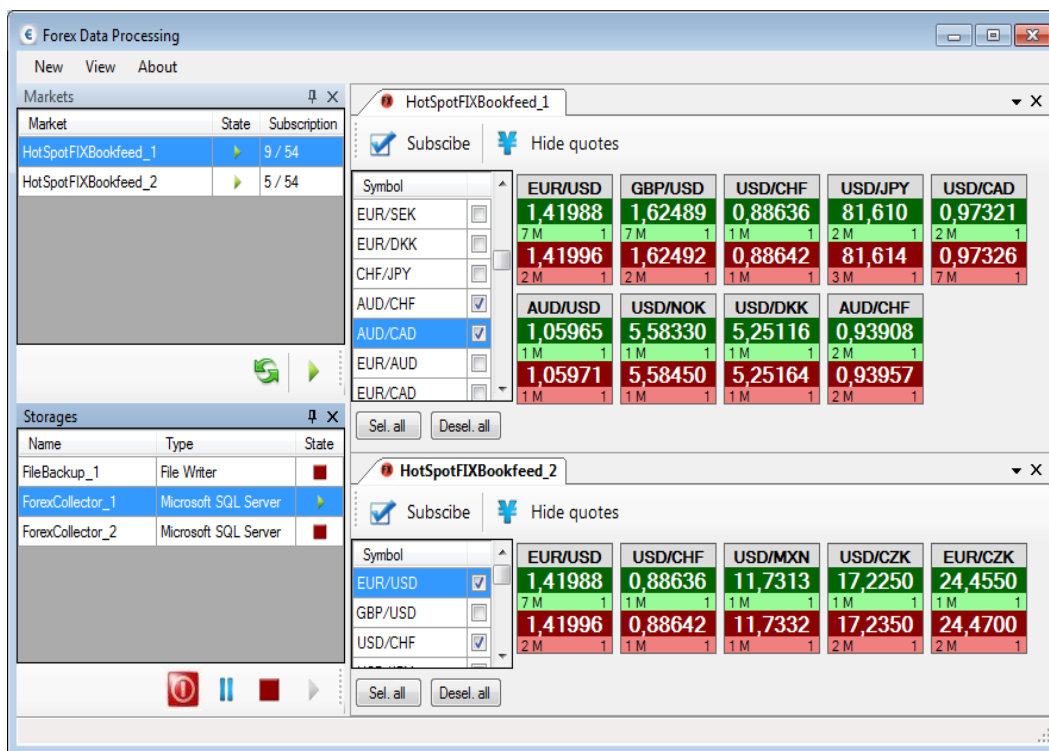
## 6.5 Grafické užívateľské rozhranie

Grafické užívateľské rozhranie je realizované pomocou Windows Forms. Pre lepšiu možnosť prispôsobenia vzhľadu podľa individuálnych požiadaviek užívateľa som využil komponentu DockPanel<sup>16</sup>, ktorá umožňuje variabilné usporiadanie okien.

Užívateľské rozhranie je centrálné pre všetky implementované knižnice. Pomocou jednej výslednej aplikácie sme schopný ovládať zber dát, ich uloženie a následne nad zozbieranými dátami vykonať filtráciu.

<sup>16</sup> <http://dockpanelsuite.sourceforge.net/>

Ukážku výslednej aplikácie zachycuje obrázok Obrázok 6.4, na ktorom vidíme dokovateľné usporiadanie okien DockPanel. Užívateľ si môže okná ľubovoľne presúvať, pripnúť na ľavý, pravý, horný alebo dolný okraj obrazovky. Okno je možné umiestniť aj na stred hlavného okna, kedy sa rozťahne na ostávajúci priestor pre efektívne využitie poskytnutého miesta obrazovky. K dispozícii je plávajúci režim okna, ktorým ho vyjmem z dokovania a umiestnime na ľubovoľné miesto obrazovky.



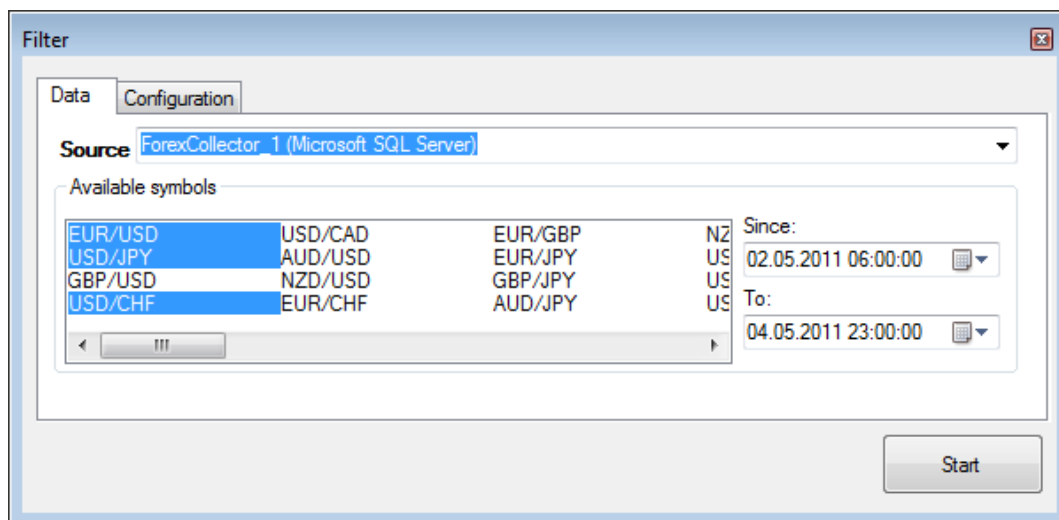
Obrázok 6.4: Ukážka bežiackej aplikácie súčasného zberu dát z 2 trhov HotSpot a zápisu do databázy

Zber dát je ovládaný pomocou okna vľavo hore nazvanom *Markets*, ktorý obsahuje načítané trhy z konfiguračného súboru. Tlačidlom spustiť otvoríme detailné okno s možnosťami trhu (*HotSpotFIXBookfeed\_1* a *HotSpotFIXBookfeed\_2*), v ktorých nakonfigurujeme odber dát. Po výbere požadovaných symbolov tlačidlom *Subscribe* spustíme odber. Vpravo od dostupných symbolov so zobrazia aktuálne kóty prichádzajúce z trhu.

Týmto však dáta z trhu len prijímame, ich ukladanie je nutné explicitne spustiť. Vľavo dole v okne *Storages* sú zobrazené načítané úložiská z konfigurácie. Ukladanie je samozrejme možné spustiť súbežne do viacerých úložísk. U jednotlivých úložísk je možné zber spustiť, pozastaviť, zastaviť a reštartovať.

Ovládanie filtra forex dát je vyčlenené do samostatného okna. V mojej implementácii je filter schopný načítavať uložené tiky s Microsoft SQL Servera. Pri výbere je možné zvoliť viacero menových párov súčasne a je potrebné vybrať časový úsek filtrovaných tikov (Obrázok 6.5). V konfigurácii nastavíme diskutované parametre. Po spustení zadáme výstupný súbor, do ktorého budú filtrované dáta uložené. Filtračný proces zaberie určitý čas o jeho priebehu informujú aktuálne

zobrazované štatistiky prijatých a zamietnutých kót. Po ukončení filtračného procesu je zobrazená záverečná štatistická správa (Obrázok 6.6).



Obrázok 6.5: Výber symbolov pre filtráciu a časové vymedzenie

Filter completed

Statistics

Start: 22. 5. 2011 21:48

Finish: 22. 5. 2011 21:48

Elapsed time: 0h 0m 21s

Symbol	Total qts.	Rejected qts.	Accepted qts.	Rejected %	Accepted %
USD/CAD	24069	963	23106	4%	96%
GBP/USD	28452	1757	26695	6,18%	93,82%
AUD/USD	24120	512	23608	2,12%	97,88%
EUR/USD	31153	2579	28574	8,28%	91,72%
USD/JPY	25373	1181	24192	4,65%	95,35%
NZD/USD	20848	1635	19213	7,84%	92,16%

Obrázok 6.6: Štatistická správa ukončeného filtrovania

Súčasťou grafického rozhrania je komponent s názvom Logger, ktorý zachytáva vznikajúce chyby a prehľadne ich vypisuje do stromu.

Implementované užívateľské rozhranie umožňuje pohodlne využívať implementovaných knižníc, avšak nijako knižnice ostávajú nezávislé a je ich možné zakomponovať do vlastných projektov.

## 7 Výsledky práce

V poslednej kapitole sa venujem zhodnoteniu dosiahnutých výsledkov. Stručne je popísaná implementovaná funkčnosť zberu a uloženia dát. Obsiahlejšie sa venujem overeniu a prezentácii dosiahnutých výsledkov filtrovania forex dát. Na záver prezentujem možné pokračovanie mojej práce.

### 7.1 Zber a uloženie obchodných dát

V časti práce venujúcej sa zberu dát som implementoval zber dát z finančného trhu spoločnosti HotSpot. Knižnica zberu umožňuje zber z viacerých trhov súčasne, jednotlivé trhy sú implementované formou zásuvných modulov, ktoré vytvárajú abstrakciu nad konkrétnou implementáciou finančného trhu. Týmto som dosiahol možnosti realizovať zber dát z technologicky odlišných trhov pomocou jednej knižnice. Pre testovacie účely som mal k dispozícii 2 prístupy k trhu HotSpot, čo mi umožnilo testovať paralelný zber dát. Prístup k trhu je platenou službou, preto prístupové údaje nie sú súčasťou odovzdanej práce.

Knižnica uloženia dát realizuje zápis zberaných tikov do úložísk. Podpora úložísk je riešená opäť formou zásuvných modulov, čo umožňuje implementovať technologicky odlišné úložiská. Produktom práce sú 2 moduly: modul pre uloženie dát na Microsoft SQL Server databázu a modul pre zápis do súboru. Navyše som implementoval synchronizáciu databáz pomocou služby replikácia poskytovanej produktom Microsoft SQL Server.

Pozbierané a uložené dáta slúžia zároveň ako vstupné dáta pre filter, ktorého návrh a implementácia bola primárnou úlohou diplomovej práce. Z objemových dôvodov nie je možné rozsiahle nazberané dáta prezentovať v technickej správe, pretože sa jedná o milióny tikov denne.

### 7.2 Filtrácia obchodných dát

Ťažiskom diplomovej práce bol teda proces čistenia dát. V kapitole 5 je popísaný matematický model filtra, ktorý som v mojej práci implementoval. Rozsiahle výstupy filtra nie je možné zahrnúť do tejto technickej správy, preto výsledky prezentujem formou implementačne zaujímavých pasáží.

#### 7.2.1 Ukážkové situácie

V tejto časti prezentujem vybrané filtračné problémy a výsledok ich riešenia dosiahnutý implementovaným filtrom.

### 7.2.1.1 Odľahlá hodnota

Filtrácia odľahlej hodnoty je základnou schopnosťou implementovaného filtra. Tabuľka 7.1 znázorňuje vyfiltrovanú chybnú hodnotu menového páru EUR/USD. Chybná hodnota je len na strane ask ceny, čo tiež znamená výrazné zväčšenie rozpätia. Preto správne kóty boli odmietnuté skalárnym filtrom ask a skalárnym filtrom rozpätia.

Bid	Ask	Čas	C <sub>bid</sub>	C <sub>ask</sub>	C <sub>rozpätie</sub>	C <sub>výsledná</sub>
1,41755	1,41760	20110517 08:04:46.8470	0,997	0,996	0,995	<b>0,995</b>
1,41755	1,41758	20110517 08:04:47.1900	0,996	0,996	0,995	<b>0,995</b>
1,41746	1,41758	20110517 08:04:47.8000	0,996	0,995	0,992	<b>0,992</b>
1,41746	1,41747	20110517 08:04:48.2070	0,996	0,994	0,991	<b>0,991</b>
1,41746	1,41747	20110517 08:04:48.4700	0,996	0,995	0,991	<b>0,991</b>
1,41744	1,41747	20110517 08:04:48.8170	0,996	0,995	0,992	<b>0,992</b>
1,41744	1,41746	20110517 08:04:49.3000	0,996	0,996	0,992	<b>0,992</b>
1,41744	1,41746	20110517 08:04:49.5170	0,997	0,996	0,992	<b>0,992</b>
1,41744	1,41746	20110517 08:04:49.7670	0,997	0,996	0,993	<b>0,993</b>
1,41744	<b>1,41842</b>	<b>20110517 08:04:50.1270</b>	<b>0,997</b>	<b>0,000</b>	<b>0,000</b>	<b>0,000</b>
1,41744	1,41747	20110517 08:04:50.3470	0,996	0,996	0,993	<b>0,993</b>
1,41744	1,41747	20110517 08:04:50.4270	0,996	0,996	0,993	<b>0,993</b>
1,41744	1,41747	20110517 08:04:50.4570	0,997	0,996	0,993	<b>0,993</b>

Tabuľka 7.1: Filtrácia odľahlej hodnoty menového páru EUR/USD

### 7.2.1.2 Dátová medzera

Filter berie v úvahu časovú os, čoho názorným výsledkom je Tabuľka 7.2. Filtrovanie je v real-time móde, v historickom by vzniknutú situáciu ošetrila aj detekcia skokov. Medzi tretím a štvrtým tikom je veľká dátová medzera a veľký rozdiel hodnôt, na čo filter zareaguje korektne a kóte s poradovým číslom nastaví neutrálnu vierohodnosť (0,5), ktorá je zároveň maximálnou povolenou hodnotou prahu nastavenej vierohodnosti.

Por. číslo	Bid	Ask	Čas	C <sub>bid</sub>	C <sub>ask</sub>	C <sub>rozpätie</sub>	C <sub>výsledná</sub>
1	80,753	80,759	20110516 09:37:56.8200	0,672	0,680	0,925	<b>0,672</b>
2	80,757	80,761	20110516 09:37:58.1800	0,505	0,510	0,855	<b>0,505</b>
3	80,754	80,758	20110516 09:37:59.0530	0,513	0,517	0,860	<b>0,513</b>
4	80,896	80,899	20110516 10:42:07.4530	0,500	0,500	0,847	<b>0,500</b>
5	80,896	80,899	20110516 10:42:09.6870	0,501	0,502	0,848	<b>0,501</b>
6	80,893	80,896	20110516 10:42:17.3130	0,500	0,500	0,847	<b>0,500</b>

Tabuľka 7.2: Reakcia real-time filtra na dátovú medzeru

### 7.2.1.3 Historický a real-time mód

Nasledovný príklad demonštruje rozdiel medzi implementovanými módmi filtra. Tabuľka 7.3 znázorňuje výsledky filtrovania zhodnej množiny tikov pomocou historického a real-time módu

filtrovania. Rozdiel medzi týmito módmi je vidieť u kóty s poradovým číslom 4. Je to miesto kde nastal potvrdený skok, avšak vierohodnosť bola v historickom móde spätne modifikovaná. Rozdiel sa prehĺbuje u kóty s číslom 5, kedy prišlo k výraznejšej spätnej úprave ako pri predchádzajúcej kóte. Do istej miery je to spôsobené aj zväčšeným rozpätím u kóty s číslom 4.

Por. číslo	Bid	Ask	Čas	C <sub>výsledná</sub> – historický	C <sub>výsledná</sub> – real-time
1	1,48214	1,48216	20110428 09:39:29.7830	<b>0,605</b>	<b>0,583</b>
2	1,48214	1,48215	20110428 09:39:30.9230	<b>0,538</b>	<b>0,533</b>
3	1,48204	1,48205	20110428 09:39:33.0500	<b>0,528</b>	<b>0,506</b>
4	1,48195	1,48203	20110428 09:39:34.4100	<b>0,599</b>	<b>0,514</b>
5	1,48194	1,48198	20110428 09:39:35.3000	<b>0,789</b>	<b>0,551</b>
6	1,48194	1,48195	20110428 09:39:35.6900	<b>0,777</b>	<b>0,708</b>
7	1,48193	1,48195	20110428 09:39:36.4570	<b>0,673</b>	<b>0,646</b>

Tabuľka 7.3: Porovnanie historického a real-time módu

### 7.2.2 Výsledky čistenia vybranej vzorky dát

Ako vzorku dát som zvolil dáta z dňa 28.4.2011 v čase od 10:00 do 14:00. Dáta pochádzajú z testovacieho zdroja spoločnosti HotSpot určeného pre vývoj systémov a obsahujú väčšie množstvo chýb. Reálne obchodné dáta by chýb mali obsahovať menej. Výsledky pre rôzne stupne prahovej vierohodnosti prezentuje nasledovná tabuľka:

Symbol	Celkom kót	Odmietnuté kóty v %		
		C = 0, 10	C = 0, 25	C = 0, 40
EUR/USD	10267	4,76%	5,66%	7,98%
USD/CHF	9265	6,22%	8,99%	12,05%
EUR/GBP	9475	6,93%	12,22%	17,71%
USD/JPY	9604	4,97%	6,28%	9,24%
NZD/USD	9527	4,35%	6,67%	10,19%
AUD/USD	9480	2,32%	3,54%	5,38%
USD/CAD	9464	4,33%	6,88%	10,22%

Tabuľka 7.4: Percentuálne zastúpenie odmietnutých kót na vybranej vzorke dát

## 7.3 Čo ďalej?

Diplomová práca spracúva široko rozsiahlu problematiku, preto nebolo možné sa všetkým častiam diplomovej práce do väčšej hĺbky. Avšak počas jej riešenia som objavil ďalšie výzvy, ktorým by sa dalo v budúcnosti venovať. Spomeniem rozšírenú variantu filtračného algoritmu a zber kompletnej knihy objednávok.



### **7.3.1 Multivariantné filtrovanie**

Použitie univariantné filtrovanie má jednu nevýhodu, ktorá limituje presnosť výsledných rozhodnutí. Pri univariantnom filtrovaní totiž sledujeme filtrované menové páry oddelene, jeden o druhom nemajú žiadne informácie. Tuna nastáva hlavný problém, pretože jednotlivé menové páry sú medzi sebou poprepájané rôznymi závislosťami, ktoré sú veľmi zložité a môžu sa postupom času meniť. Na skúmanie vzájomných vzťahov menových párov by sa dali aplikovať prostriedky dolovania dát. Takto vydolované závislosti by sme použili pri určovaní vierohodnosti prichádzajúcich kót. Vhodným príkladom využitia je cenový skok, kedy by sme pomocou závislostí skúmali menové páry, ktoré sú so skúmaným symbolom závislé. Ak by prišlo k cenovému skoku aj u sledovanej, je to impulz pre potvrdenie skoku. Opačný prípad je znamením „falošného poplachu“.

Toto rozšírenie je veľmi rozsiahle z dôvodu veľkej rozmanitosti vznikajúcich závislostí. Možným riešením je závislosti vydolovať z existujúcich dát. Oveľa sofistikovanejším riešením je ich adaptívne dolovanie v priebehu filtrovanie. Túto problematiku si dokážem predstaviť ako tému celej odbornej práce.

### **7.3.2 Zber a filtrácia úplnej knihy objednávok**

Obchodné modely obvykle pracujú len s vrchom knihy objednávok. Existujú aj sofistikovanejšie, ktoré pracujú s kompletnou knihou. Zber úplnej knihy je relatívne ľahko realizovateľný. Filtrovanie úplnej knihy naopak bude oveľa zložitejšie a bolo by možné sa mu venovať do hĺbky v ďalšej práci.

## 8 Záver

V úvodnej časti diplomovej práce som zhrnul poznatky nadobudnuté počas štúdia oblasti finančných trhov. Stručne sú popísané všeobecné princípy obchodovania a následne je zacielená pozornosť na obchodovanie na devízovom trhu. Detailnejšie som sa venoval obchodným dátam. Tematicky by sa práca dala rozčleniť do 3 celkov.

V časti venovanej zberu dát som implementoval funkčný prototyp knižnice komunikujúci s finančným trhom podľa protokolu FIX. Knižnica je formou zásuvných modulov rozširiteľná o ďalšie trhy.

Na zber dát nadväzuje ich uloženie, ktoré je implementované ako samostatná modulárne rozširiteľná knižnica. Navyše som načrtol a implementoval redundantné uloženie dát, možnosti ich synchronizácie medzi databázovými servermi.

Primárne som sa venoval filtrácii dát. Na základe odbornej literatúry som analyzoval možné chyby vo vysoko-frekvenčných dátach, navrhol a implementoval adaptabilný filter obchodných dát.

Problematika finančných trhov je v súčasnosti relatívne nová, veľmi živá a prebiehajú rozsiahle výskumy. Veľký dôraz sa kladie na využitie veľkého potenciálu výpočtovej techniky a možností algoritmickej obchodných stratégií. Téma diplomovej práce bola zaujímavá z hľadiska nadobudnutia poznatkov z oblasti finančných trhov. Taktiež som sa zoznámil s možnosťami filtrovania časových radov. Z praktického hľadiska som si zdokonalil analytické, návrhárske a implementačné zručnosti. Práca bola pre mňa veľmi prínosná, načrtol som možnosti ďalšieho smerovania práce, čím sa otvára priestor pre nadväzujúcu prácu.

# Literatúra

- [1] Durbin, M.: All About High-Frequency Trading, McGraw-Hill, 2010, ISBN 978-0071743440.
- [2] Galant, M., Dolan, B.: Currency trading for dummies, Wiley Publishing, 2007, ISBN 978-0-470-25143-0.
- [3] Gencay, R., Dacorogna, M., Muller, U.,A., Pictet, O., Olsen, R.: An Introduction to High-Frequency Finance, Academic Press, 2001, ISBN 978-0122796715.
- [4] Johnson, B.: Algorithmic Trading and DMA: An introduction to direct access trading strategies, Wiley 2010, ISBN 978-0956399205.
- [5] Obchodujeme FOREX I - Financnik.cz [online][citované 7.1.2011] Dostupné na URL: <http://www.financnik.cz/komodity/zkusenosti/obchodujeme-forex-1.html>
- [6] Tsay, R., S.: Analysis of Financial Time Series (Wiley Series in Probability and Statistics), Wiley, 2010, ISBN 978-0470414354.
- [7] Luboslav, L.: Datové sklady, analýza OLAP a dolování dat, Computer Press, 2003, ISBN 80-7226-969-0
- [8] .NET Framework overview [online][citované 5.5.2011] Dostupné na URL: <http://www.microsoft.com/net/overview.aspx>
- [9] SQL Server overview [online][citované 5.5.2011] Dostupné na URL: [http://msdn.microsoft.com/en-us/library/ms166352\(v=sql.90\).aspx](http://msdn.microsoft.com/en-us/library/ms166352(v=sql.90).aspx)
- [10] FIX Protocol [online][citované 9.5.2011] Dostupné na URL: <http://www.fixprotocol.org/>
- [11] FIX Specification for MarketData [online][citované 10.5.2011] Dostupné na URL: [http://www.hotspotfx.com/pdfs/Bookfeed\\_Spec.pdf](http://www.hotspotfx.com/pdfs/Bookfeed_Spec.pdf)
- [12] SQL Server Replication overview [online][citované 12.5.2011] Dostupné na URL: <http://msdn.microsoft.com/en-us/library/ms151198.aspx>
- [13] Peer-to-peer transactional replication [online][citované 12.5.2011] Dostupné na URL: <http://msdn.microsoft.com/en-us/library/ms151196.aspx>

# **Zoznam príloh**

Príloha A: Obsah CD